1-2021

Towards Effective Wireless Intrusion Detection using AWID Dataset

Dino Linekar Robert Wilson
dxr4536@rit.edu

Towards Effective Wireless Intrusion Detection using AWID Dataset

by

Dino Linekar Robert Wilson

A Thesis Submitted in Partial Fulfilment of the Requirements for the Degree of Master of Science in Networking and System Administration

Department of Computing Sciences

Rochester Institute of Technology

RIT Dubai

January 2021

Committee Approval

Dr Ali Raza — Date
Professor of Computing Sciences
Thesis Advisor

Dr Muhieddin Amer — Date
Professor and Chair
Electrical Engineering and Computing Sciences

Abstract

In the field of network security, intrusion detection system plays a vital role in the procedure of applying machine learning (ML) techniques with the dataset. This study is an IDS related in machine, developed the literature by utilizing AWID dataset. There tends to be a need in balancing a dataset and its existing approaches from the analysis of its respective works. A taxonomy of balancing technique was introduced due to the lack of treatment of imbalance. This attempt has provided a proper structure defined on all levels and a hierarchical group was formed with the collected papers. This describes a comparative study on the proposed or treated aspects. The main aspect from the surveyed papers were found that: understanding of the existing taxonomies were not in detail and there were no treatment of imbalance for the utilized dataset. So, this study concludes a gathered information in these aspects. Regardless, there are factors or weakness have been seen in any adaptations of the intrusion detection system. In this context, there are few findings that are multifold with contributions. Thus, to best of our knowledge, the study provides an integration with the observation of threshold limit and feature drop selection method by random samples. Thus, the work contributes a better understanding towards imbalanced techniques from the literature surveyed. Hence, this research would benefit for the development of IDS using ML.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Wireless networks are being considered as the most convenient and unavoidable in daily life. The 802.11 networks, are also referred to as Wi-Fi which are the popular choice of low cost wireless connectivity. It allows a quick setup in an enterprise environment for the exchange of data with standards providing security. The 802.11i document, provides the specification for security (Wi-Fi) [1]**.** It is known that, the usage of internet has led to an enormous information boom. This massive expansion had lead the network to become vulnerable due to the open standards which are available.

These huge volumes of data had become a challenge to address, process and store. As this makes the attackers to sneak into the network easily and target in dispatching the private information. Regardless, the utilization of previous several security applications could also become a victim. To secure these data, many organization and experts are involved in this development. The development occurring in cybersecurity are becoming vigorous and has pulled insignificant attention globally. There are consistent explorations in deploying and developing a novel intelligent security system that can manage and withstand against the intrusion events.

Such type of external mechanism are known as Intrusion Detection System (IDS). This system helps in identifying and reacting for an intrusion event in a timely fashion [2]. The network traffic is monitored to detect whether the traffic is normal or malicious. The variety of security attacks on Wi-Fi makes it as a high interest and a trending area of research. So, popular intrusion detection techniques have been applied to wireless networks [3]. This has become an emerging area of research with the advent of Machine Learning (ML).

Continuous focus has been committed for the development of the datasets that depends upon ML techniques [4]. Despite the progress, there is also a basic issue of imbalance of the datasets. This brings out to be a bias and but achieves the accuracy needed [5].

There are researches that involves balancing techniques as its major findings. In [6], the work reviews the impact of the imbalance class distribution and introduced a computational system which comprises the arrangements for both data and algorithm levels. Similarly, an overview of existing methodologies for classification of imbalanced dataset are discussed in [7]. Thus, handling the issues in class imbalance is more important from the understanding

It is well known that, there are literatures addressing the issues of an intrusion detection system in machine learning. But, these works unclear and incomplete with some of the factors, due to the lack in identification or experience to a certain extent. Thus, it is essential to support and contribute the development and research group on the predominant results performed by utilizing the imbalance dataset in cybersecurity.

1.1 Motivation

Network security has become a basic general issue all over. Considering the rate of cyber-attacks, their drastic growth, strategies and advancement are now capable for attacks. So, Intrusion Detection Systems (IDS) are one among the solutions that have proven against these attacks. Although, the Wi-Fi requires a good knowledge of its difficulties and limitations on implementing an IDS system. Hence, the ML based IDS exhibits an efficient and successful performance with the imbalanced dataset. Here, the network traffic data of imbalanced class distribution are interacted several ML classifier algorithms to execute a balanced and enhanced output. This context leads the path for researchers to study and investigate on the area of biased or imbalance class distribution.

There are several development groups that perceives the significance to have a balanced dataset and focusing on reduced bias in machine learning. The work presented in [8] and [9], introduces that when the dataset comprised of more instances with one class than the attack class, a low detection rate can be observed with the presence of imbalanced data. A downside of this in [9], examines the performance towards solving, either by loss of data or overfitting. It expects the majority class to be biased in order to recognize all the classes. Hence, by then, it is now easy to address the issues by having a spotlight on balancing of dataset [5].

Hence, in previous works in wireless intrusion detection, the machine learning algorithms with high accuracy justifies an enhanced approach by using its adversaries and tools which are considered in attack networks

The goal of this examination is to use different strategies in bringing up an IDS which help to propel the imbalance data classification.

1.2 Contributions

The major contribution for this study includes the following:

- A comprehensive work among different authors that targeted on balancing the datasets and its approaches were observed.
- Analyzing and investigation was carried on the Machine Learning techniques used across the collected papers.
- Exploration of the labelled dataset of wireless network. This work was focused only in the domain of Supervised Machine Learning.
- Improving the performance metrics of the Machine Learning techniques which are associated.
- Implementation of the required algorithms for evaluating the technique which is proposed.
- Providing the source code guideline and compare & contrast the results of the new improvements.

1.3 Thesis Structure

The rest of the thesis is as follows:

Chapter II, provides a literature review related to AWID dataset.
Chapter III, discusses the class imbalance and a structure of proposed taxonomy.
Chapter IV, highlights the implementation and initial setup procedures.
Chapter V, evaluation of the AWID dataset in various phases are provided with results.
Chapter VI, concludes the thesis with the direction of the contribution provided.

Chapter 2

Related Works

This section provides the discussion of the relative work proposed and literatures treated to balance the utilized AWID dataset. This discussion is as follows:

2.1 Relative work with AWID

An agent-based malicious detection framework was proposed by the author in [10]. The intrusion activities are detected during the process with the use of Artificial Neural Network (ANN). In this work, the experimentation are carried out specifically on the AWID-CLS-R subset to characterize every instances as a normal or an attack. It has been demonstrated that the proposed framework on AWID-CLS-R subset has provided an exceptionally precise results having 99.3%.

In [11], the AWID-CLS-R subset was used a multi-class classification for the experiment. As a result, the author has used deep learning approach for achieving an improvement in an overall accuracy to 98.67%.

In [12], the creator has performed the attack classification on AWID-CLS-R subset by applying eight traditional supervised machine learning classifiers. In order to train the classifiers, the author has integrated 20 features as a combined one and also the feature selections done manually. AdaBoost, OneR, J48, Naïve Bayes, Random Forest, ZeroR and Random Tree were the algorithms utilized to observe the attack classification performance. The work has produced an overall accuracy from 89.43% to 96.2%.

A framework in [13] was proposed, which is used to detect active attacks by using Stack Auto Encoder (SAE). This also tends to be an unsupervised learning approach for the feature selection process. This frame structure used the regression layer, following supervised learning technique and SoftMax activation function which resulted a highest accuracy of 97.7%. In addition, the works has also produced by highlighting the best feature among the three machine

learning methods. This work in [14, 15] was focused on AWID-CLS-R subset in order to enhance the detection in impersonation attack. Out of 4 classes of the subset, 2 classes were removed and 2 were reserved (impersonation attack class and normal traffic class). Using Artificial Neural Network (ANN) for attack classification and utilizing the Decision Tree, Support Vector Machine (SVM) for the approach had a precise outcome of 99.86% to detect the impersonation attacks.

The author in [16, 17] considered the reduced classification version of CLS and ATK class subsets by applying five supervised machine leaning classifier algorithms. The AdaBoost, Random Forest, Random Tree, OneR and J48 were the algorithms utilized in this work. Before the application, the features were evaluated and used Information Gain and Chi-Square measures for ranking, Based on this evaluation, the classifiers were applied to the respective subsets which then resulted as the highest accuracies with 41 features. The outcomes shows that the Random Tree classifier on AWID-CLS-R gained 95.12% and Random Forest on AWID-ATK-R gained 94.97%. It is also noted that, reduce in features to a certain limit results in improving the accuracy.

A research work in [18] proposed a distributed network intrusion detection system named TermID. This framework was developed to improve efficiency, without the exchange of sensitive data. The Classification Rule Induction (CRI) and Swarm Intelligence Optimization (SIP) were utilized in accomplishing a productive model. It has two operational units: (i) monitor node and (ii) central node. The AWID-ATK-R subset was considered and physically separated for each nodes. But it is noted that, the creator did not publish the accuracy.

An ensemble learning algorithm approach in [19], have used AWID-CLS-R dataset for the multi-class classification. The author was able to achieve an accuracy of 95.88% from this work. There was also another observation when the attack classes are combined into one. This resulted around 99.11% of accuracy, recorded from the observation. But, the impersonation and injection attacks caused a very low accuracy. So, a new machine learning model was applied to distinguish each of these attack classes.

A framework in [20] was proposed for a classification to differentiate a complex sample from the easier one. This framework with Wireless Network Intrusion Detection System (WIDS)

used deep learning approach to achieve 98.54% for multi-class classification and 99.52% for binary class classification.

Most of the work listed above have certain machine learning techniques that played an important role respectively. And, thus, their use of these approaches have brought out a difference in their outcome cycles. Therefore, these are some of the relative works based on the AWID dataset.

2.2 Impact of dataset balancing

The data imbalance is a typical event which are occurred in the vast majority of the real-time datasets.

When a dataset having one of its classes enormously dominating the strength of other classes, it is considered to be imbalanced. The binary classification dataset are the more salient to this situation, as they are completely imbalanced.

According to the various observations recorded, it is known that there can be any form of unequal distribution of both major and minor classes. Mostly, the observations have majority class more significant than the minority class in a dataset. These cases can be related to the application like fraudulent telephonic calls, bank transactions etc., in which the imbalance levels can be noticed easily. From the applications, it very well considered as a major preference for minor events and normal observation as major events [21].

The complications of the anomaly detection in a dataset is continuous in the real time scenario. On the account of network intrusion detection, the rate of attack packets occurred would be lower when compared to the rate of normal packets occurred. This can cause issues with regards in executing a wrong outcome and also may lead to misunderstanding that the dataset is balanced, even if it is an imbalanced one. Thus, the effort that was made towards the execution would be an overall influenced output, which is not useful [7].

When there is an imbalance in the data represented, it is very important to understand the minority and majority classes. The work in [22], inspected that the impact of class imbalance was addressed by making artificial data with several possibilities of level of imbalance and different size of the training set. This resulted with no issues or change across all levels of imbalance during the process.

There are certain domains with which intrusion happens due to low frequency and its related factors, as they lack in data. This type of cases can also be addressed to some extent in detecting the events. Thus, [22] examines the difficulties in understanding and machine learning techniques.

So, in general, until now there is nothing initiative towards any large scale research in this context. Therefore, it can be now clear that there should be a path for solving all the issues of classification of imbalance data in the future.

2.3 Taxonomy of balancing level techniques

Data Level

In [23] Data level, the nature of the class is balanced to avoid the class imbalance distribution. This approach is been used as the preprocessing step for resampling the class distribution. Over sampling and under sampling are the two main classification in sampling methods.

Over sampling and Under sampling - SMOTE

In Over sampling, a random replication of the minority class will be created and in under sampling a subset of the majority class is selected to balance the class distribution. Whereas, another popular oversampling method called SMOTE, successfully avoids overfitting when new minority data are generated [23].

Algorithm level

In algorithm level, the training data distribution is not modified while handling the class imbalance. They are combined into an overall approach to address the class imbalance problem. It actually provides a good classification performance for big data [22].

Ensemble method

Ensemble learning method [24] is a combination of several models in improving machine learning results. This approach allows the process by providing a better performance than the individual classifier used. In this method, bagging and boosting are the most commonly used approaches.

Cost sensitive Learning

In data mining, the objective of this approach is to limit all the convincing expenses of known classes. This cost sensitive learning is a type that considers and especially takes on misclassification costs. Cost insensitive learning is another category which is different as such, it approaches the diverse misclassification in an unexpected way. It doesn't consider the misclassification costs. And, this cost sensitive learning is the most commonly used approach in solving the class imbalance [25].

Hybrid level

These hybrid method combines or integrate various machine learning models into its approach. In this method, since it uses different models as input, there will be better performance compared to an individual. It helps to exploit different mechanisms of the basic model and reduce its limitation [26].

2.4 Literature of balancing AWID dataset

This section, presents the literatures that discuss the treatment or proposed concept related with the AWID dataset.

A frame work is proposed by the authors in [27], where Synthetic Minority Oversampling Technique (SMOTE) is used to overcome the imbalance problem in the AWID dataset. This framework with SMOTE is an intelligent over-sampling technique, which is used to balance the AWID-CLS datasets (i.e.) samples are added to the minority classes to attain equal distribution among the classes.

In [28], a novel intrusion detection framework has been proposed based on feature selection and ensemble learning techniques. First, CFS-BA algorithm is proposed for dimensionality reduction. It aims to select the optimal subset, based on the correlation between the features. Then an ensemble of C4.5, Random Forest (RF), and Forest by Penalizing Attributes (Forest PA) classifiers is developed as an approach to produce the classification model. Lastly, voting techniques have been used on average of probability distribution. The outcome produced using the three (AWID, NSL-KDD and CIC-IDS 2017) datasets reports that, the CFS-BA Ensemble method performed better compared to other approaches.

The authors in [29], have adopted the same recorded AWID dataset samples from other literatures used. They have applied few preprocessing steps to get a modified dataset in order to maintain the 802.11 fields and its verbosity just like the original dataset. Feature selection using Gini index method is applied after the preprocessing step. And hence the imbalance in the distribution is corrected before training using Random under sampling.

Following this, in [30] the AWID training set is resampled to balance the dataset in order to (i) create a balanced training set and (ii) reduce the size of the original training set. These training sets were dramatically reduced by using Random under-sampling, Random over-sampling or SMOTE techniques before proceeding to the classification step.

An investigation study carried out in [31], to design a powerful and productive intrusion detection framework. The authors propose a framework which is composed with modules like feature selection and dimensionality reduction, to handle imbalanced class distributions and classification. The correlation based subset evaluation techniques and searching algorithms are applied in feature selection mechanism while in feature dimensionality reduction, auto-encoder

and principal component analysis is applied. Hence, several classifiers and imbalanced class handling approaches are evaluated to determine the best suited one for this proposed intrusion detection framework. In this evaluation, the authors have used twelve well known classifier algorithms over four different attribute sets: 32, 10, 7, and 5 FSGs. The outcome reports that selection or rejection from the optimal attribute have produced an enhanced ~~results with~~ time processing and accuracy results.

Finally, this work analyses research published to identify methods employed to balance the AWID datasets. A novel approach will be proposed and implemented as a contribution for this thesis.

2.5 Preprocessing

The term preprocessing is an essential step followed in every machine learning experimentation. It is considered as a vital part for both the classifier model which is used and improving the overall performance of the classification. The actual concept of this component is that, the features or attributes of the dataset are now easily set to a path in order to get interpreted by the selected algorithms.

In this study, preprocessing step was needed and was carried on one of the subset of the AWID datasets. This considered subset were able to accomplish the stage of preprocessing.

Chapter 3

Analysis of related work

In this section a unique approach is used to reviewing and presenting the current literature. Due to the cross-sectional study undertaken in this work, the literature related to the datasets is presented with an analysis and cross-referenced with the literature on dataset balancing techniques.

3.1 Importance of balancing techniques

The objective of developing a dataset for intrusion detection systems is to capture normal and abnormal events which can be used to train ML algorithms for classification purposes. Abnormal events or anomalies may be caused either due to the poor performance of software or due to malicious attacks in a network. In a well-designed software and well protected network services, the ratio of normal traffic to abnormal traffic (anomalies) is expected to be high. Developing datasets which are real-world and not synthetic or even semi-synthetic means that data will inevitably be imbalanced. Abnormal events are not that common; however, their impact can be significant. When preparing a dataset, enough of these abnormal event samples must be available to remove any bias when training a ML algorithm [32].

Much of the literature on ML methods applied to these datasets seems to have been based on reporting the accuracy of an algorithm. The research to date has been predominantly in optimizing the algorithms used in ML, towards improving the metrics shown in Table 1. The papers surveyed in this study do not convincingly show that imbalance datasets are treated and to what degree.

Table 1. Confusion Matrix

Confusion Matrix		**Classification**	
		Positive	**Negative**
Target	**Positive**	TP	FN
	Negative	FP	TN

Confusion Matrix is a performance measurement for machine learning classification. The confusion matrix in Table 1, comprises of four items for binary classifiers:

True Positives (TP) - when the classifier identifies the true positive label as positive
True Negatives (TN) - when the classifier identifies the true negative label as negative
False Positives (FP) - when the classifier identifies the true negative label as positive
False Negatives (FN) - when the classifier identifies the true positive label as negative

In the context of cybersecurity research, a well-known understanding is that a positive event is defined as a malicious event and the correct classification of such an event is deemed as a true positive outcome. A negative event is a benign event and the correct classification is deemed as true negative. Inaccurate classification can mean that a benign event is classified as a malicious event. This misclassification is deemed as a false positive. Likewise for a malicious event to be classified as a benign event is deemed a false negative [33].

Table 2. Class Distribution for AWID-CLS-F-Trn

Class Label	Count
Normal	157749037
Impersonation	1884378
Injection	1530373
Flooding	1211459
Total	**162375247**

In table 2, the class distribution for high-level labelling method (CLS) in AWID for various attacks is derived from [34]. Impersonation, Injection and Flooding are the three types of malicious events in the full set dataset. Figure 3 displays the breakdown of imbalanced class distribution for AWID-CLS-F training dataset [34]. A similar exponential trend observed in Figure 1 and 2, can also be noticed here.

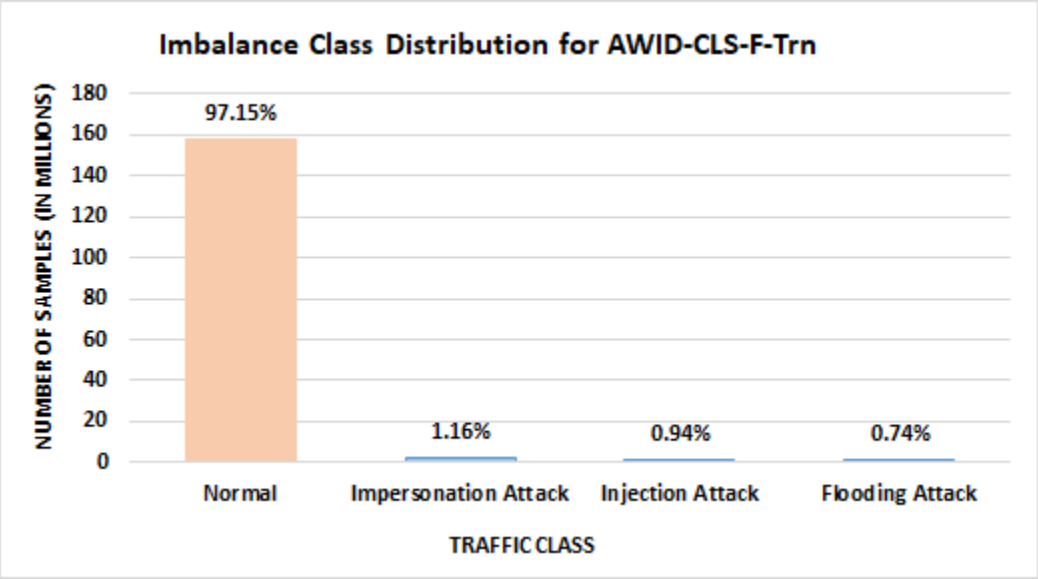


Figure 1. Imbalance Class Distribution for AWID-CLS-F-Trn

Table 3: Imbalanced Ratio Distribution for AWID-CLS-F-Trn

Normal Traffic Count	**AttackTraffic Type**	**Count**	**Normal to Attack Ratio**
157749037 **(97.15%)**	Impersonation	1884378 **(1.16%)**	84 ***to*** 1
	Injection	1530373 **(0.942%)**	103 ***to*** 1
	Flooding	1211459 **(0.746%)**	130 ***to*** 1

In Table 3, we provide an approximation of the ratios of benign to malicious traffic in the various minority classes from table 2 [34]. Impersonation, Injection, Flooding has a ratio of 84 *to* 1, 99 *to* 1 and 103 *to*1 respectively. The ratio of benign traffic to the sum of all malicious traffic is 9 *to* 1. Hence, a machine learning algorithm is susceptible to yield a high False Negative rate and a low False Positive rate.

3.2 Review of contentions in existing taxonomies

Perhaps the most comprehensive account of existing techniques and an indication of a taxonomy is provided in [32], [6] and [7]. The authors in [6] review the approaches which span over the last 8 years. In contrast, [32] and [7] do not specify the year span of their reviews.

Table 4 derives the taxonomy based on approaches in [32], [6] and [7]. Two important classifications emerge from the studies in [32], [6] and [7]: techniques classed as data level; techniques classed as algorithm level. Collectively, these studies converge on the definition of data level methods to include data sampling and feature selection approaches, while algorithm level methods include cost-sensitive and hybrid/ensemble approaches.

In [6] and [7] the authors define data level methods to include data sampling and feature selection approaches, while algorithm level methods are defined to include cost-sensitive and hybrid/ensemble approaches. Across all three surveys shown in Table 4, several divergent accounts of algorithm level classifications have been proposed, creating numerous discrepancies. In [32] the major deviation is in the algorithm-level definition. In contrast to [6] and [7], subcategories of algorithm-level are not defined in [32]. The subcategory of *one class*, however, is mentioned in [32] under the discussion of algorithm-level but not distinctly classified as in [6] and [7].

Table 4. Contentions in Existing Taxonomies

Reference	Title of Paper	Publication Date	Citations	Data Level			Algorithm Level						Cost Sensitive Learning	Boosting Approaches
				Data Level	Feature Selection	Data Sampling	Algorithm Level	Cost Sensitive Learning	Ensemble	Hybrid	One Class	Improved		
[35]	A survey on addressing high-class imbalance in big data	2018	55	☑	☑	☑	☑	☑	☑	☑	☐	☐	☐	☐
[32]	Classification of Imbalanced Data: A Review	2009	788	☑	☐	☑	☑	☐	☐	☐	☑	☐	☑	☑
[7]	Classification with class imbalance problem: A review	2015	157	☑	☑	☑	☑	☑	☑	☑	☑	☑	☐	☐

The derived taxonomies in [6] and [7], which are more recent, show that feature selection is a subcategory of data level whereas [32] does not. We cautiously suggest that this may be because the feature selection approach gained popularity in the study of dataset bias after the publication on [32]. Both [6] and [7] discuss techniques in feature selection such as principal

component analysis (PCA) and the likes since the publication of [32]. The specifics of the feature reduction techniques and its development over the years is beyond the scope of this paper. In contrast to [7] which addresses the concept of *Improved learning*, [6] and [32] do not discuss this as a subcategory or part of the taxonomies provided.

In statistics and machine learning, ensemble methods use multiple learning algorithms to obtain better predictive performance than could be obtained from any of the constituent learning algorithms alone [36]. In contrast to [7] which defines ensemble as boosting (an iterative technique which adjusts the weight of an observation based on the last classification), [6] defines ensemble as both bagging (a way to decrease the variance in the prediction by generating additional data for training from dataset using combinations with repetitions to produce multi-sets of the original data) and boosting. The definition of ensemble provided in [6] aligns better with the definition provided in [35].

Another major deviation observed in [35] and [7] from [32] is the inclusion of ensemble under algorithm level in [6] and [7]. In [32] ensemble, cost sensitive and other boosting are included as subcategories of boosting. This is not shown in Table 4 due to the lack of space. The definition of ensemble provided in [32], however, agrees with the definition provided in [35].

3.3 The IDS dataset used in this study

This section provides an overview of the AWID dataset and table 10 highlights the key attributes such as domain, purpose of IDS, year of publication, volume, number of features and traffic types for all three datasets.

The most popular Aegean WiFi Intrusion Dataset (AWID) was developed and made publicly available by the Info sec lab, University of Aegean. This dataset were particularly assigned for Wireless IDS and introduced by the author in [12]. Even though there are other dataset which are universally utilized for the research in network IDS, AWID stands as a primary endeavors created from the wireless network. The AWID is comprised of subsets which are easily available in a classification format of datasets. There are no artificial traffic records within the AWID. These records were been normally delivered and observed the real traces of both normal

and intrusive traffics from a protected WLAN network with WEP security protocols. The author contend that the AWID has brought out a new significant commitment in exploring WIDS and claims it as the first publicly available dataset. This could possibly be a benefit for the various wireless networks that depends on 802.11 standards [36].

From [38], it is classified into of two equal dataset with different labeling method such as: AWID-CLS and AWID-ATK. They are labeled according to CLS representing the classes and ATK representing the actual attacks. Each of these two dataset is comprised of a full subset: AWID-CLS-F and AWID-ATK-F and a reduced subset: AWID-CLS-R and AWID-ATK-R. The reduced subset are mainly considered and utilized only at the initial stage of research and examinations. This is because, they are efficiently analyzed easily and are available in smaller size. Whereas, the full sets are bit large in size and need enhanced version of wireless IDS to make up with the large volumes of data. Additionally, every subset of AWID classification comprises two versions, they are: training and testing and they are denoted as “Trn” and ”Tst” respectively.

There are 155 attributes in total with all the AWID subsets. These attributes are dedicated as 154 among these are instances of features and 1 is a class instance representing a traffic record if it is a normal or attack.

In this thesis, among the AWID subsets, the AWID-CLS-F-Trn and AWID-CLS-R-Trn subsets are utilized for this study. The AWID-CLS-F-Trn subset is considered mainly for the overview of the taxonomy and the AWID-CLS-R-Trn subset is used for implementation and evaluation process.

3.4 Methodology towards proposed taxonomy

It can be seen from the analysis provided in Table 4 that the categorization of balancing techniques proposed is quite wide ranging. What stands out in the table is the lack of consensus on the algorithm level techniques. Opinions differ on the ensemble techniques predominantly. This further supports the idea that a consensus on taxonomy of techniques is required, which is proposed in Section IV.

Our approach to deriving a new taxonomy is based on the study of work published in the cybersecurity domain anchored on the AWID IDS datasets. The criteria used for selecting papers related to IDS and dataset balancing is specified in this section as follows:

Criteria for selecting the papers related to IDS were as follows:

- Publications were only included if they were relevant to the three datasets being studied.
- Publications were only included if certain keywords related to dataset balancing were found.
- Publications were only included if they were published between 2016 and 2020 to align with the first public announcement of the dataset.

Criteria for selecting the papers related to dataset balancing were as follows:

- Publications were only included if they were highly cited.
- Publications were only included if they were highly cited in the IDS publications. This indicated that the IDS paper recognised the importance of dataset balancing.
- Publications were only included if they were published between 2000 and 2020 to explore the advances in dataset balancing over the last two decades for this cross-sectional study.

To come up with this methodology, previous approaches published in [38] and [39] were reviewed and analyzed. The two papers were compared, in [38] more advanced techniques were proposed due to the recent advancements provided by google scholar platform reported in this paper.

From [38], the google scholar as a platform provides access to key information about the citation of a paper. The name of the primary dataset paper is first entered in the google scholar search engine. The number of times the paper has been cited is displayed and clicking on it leads to the total list of cited papers. The papers are filtered down by clicking the checkbox **" Search within cited articles"**, the keyword search and custom range process used for the datasets are explained below:

The keyword **"imbalance"** was used for the AWID dataset, 23 papers were found. A custom range option is also available in google scholar to select the key papers. The range applied for our research was *2016-2020* for AWID respectively.

3.5 key findings and proposed taxonomy

The study of the published work outlined presented and discussed in this section. The evidence collected through the findings of this study are used to propose a new taxonomy for balancing techniques. Furthermore, the findings presented in this section encourage research and development of new techniques or the application of existing techniques which are discussed in the future work section.

3.5.1 Analysis of Published Work

A systematic review of the literature in the cohort of published works from Section III allowed us to divide the work into groups according to the level of contribution each work makes towards balancing of a dataset. This grouping has been done in a hierarchical order as shown in Figure 2 with the first level determining whether the paper has a contribution or not. The second level identifies the extent of the contribution in terms of a proposed method or the application of an existing method. In the case of non-contributing work, the second level identifies whether imbalance has been recognized and mentioned or not.

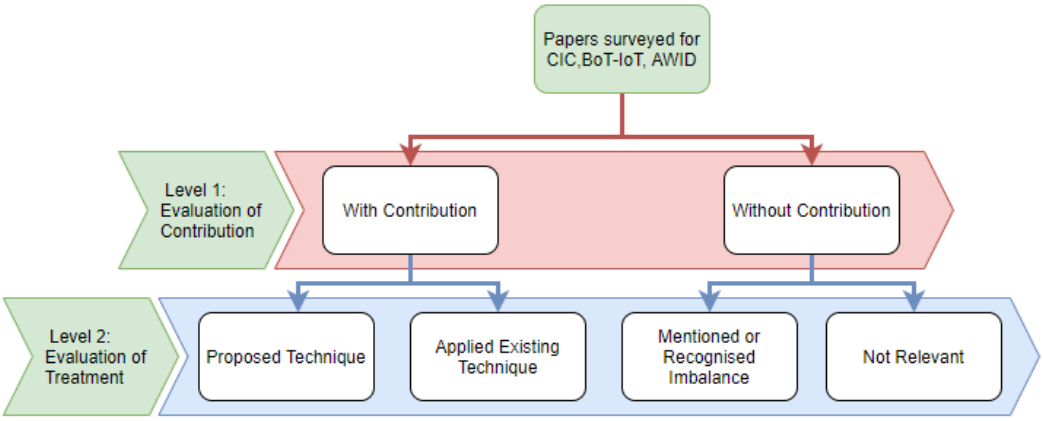


Figure 2: Hierarchical grouping of published work

A more thorough definition of the grouping has been provided below:

- **Proposed:** The authors have proposed a technique to solve imbalance.
- **Applied Existing:** The authors have applied existing techniques in solving imbalance.
- **With Contribution:** This is a cumulative of papers in which the authors have either proposed or applied existing techniques.
- **Mentioned:** These are the papers in which the authors have mentioned an imbalance technique with respect to our analysis but have not treated it.
- **Not Relevant:** The authors have mentioned imbalance in general and not with respect to our analysis of dataset imbalance.
- **Without Contribution:** The authors of these papers have either mentioned imbalance or did not have any discussion relevant to the imbalance of the datasets.

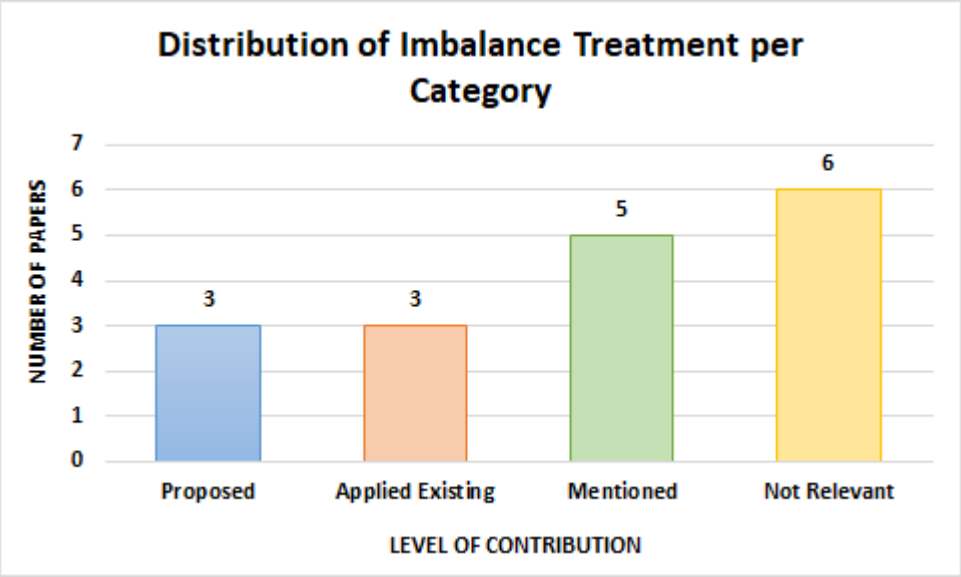


Figure 3: Comparison of the imbalance treatment categories across AWID datasets

The categories: *with contribution* and *without contribution* have been added for the ease of presenting the analysis. Papers that have proposed a new technique or applied an existing technique to deal with the imbalance are labelled as ***With Contribution***. Furthermore, papers that have no relevance to dataset imbalance or have reservedly mentioned the **word** imbalance are collectively labelled as ***Without Contribution***.

Figure 3 shows the distribution of published work across the four groups defined for AWID datasets. Figure 4 presents a cumulative percentage distribution across the four groups irrespective of the datasets used. It has been observed that only 37% of the papers have either proposed or applied techniques to treat dataset imbalance and the remaining 63% of the papers have not contributed to this study. These results further support the idea that there is a lack of attention to imbalance because 45% of the papers are ***not relevant*** which takes precedence over other groups.

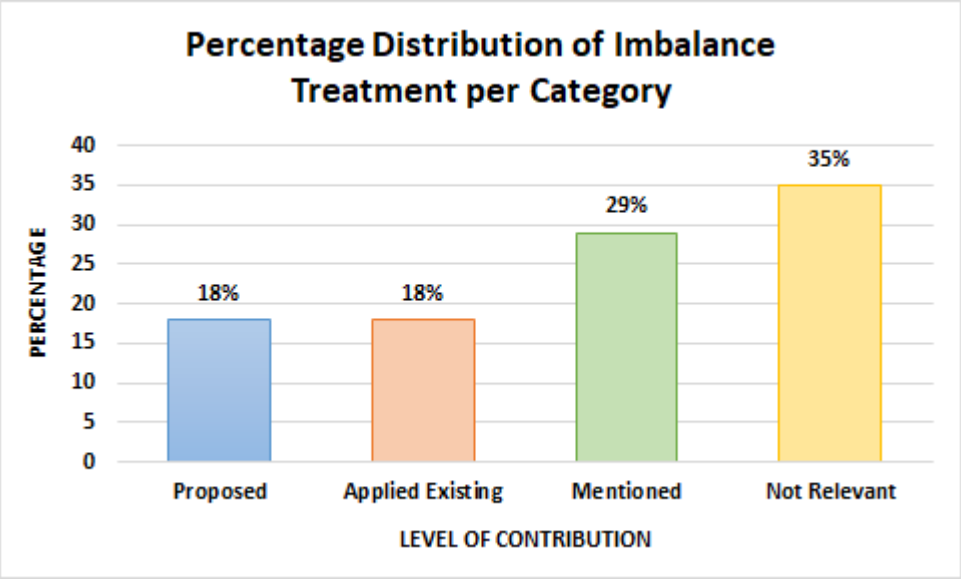


Figure 4: Categorization of Imbalance Treatment- A comparison of all papers surveyed, across all categories of imbalance treatment.

The "***without contribution"*** category shown in Figure 5 takes precedence with the highest count being for CIC. In the "***with contribution"*** category the ***applied existing*** takes precedence as shown in Figure 5.

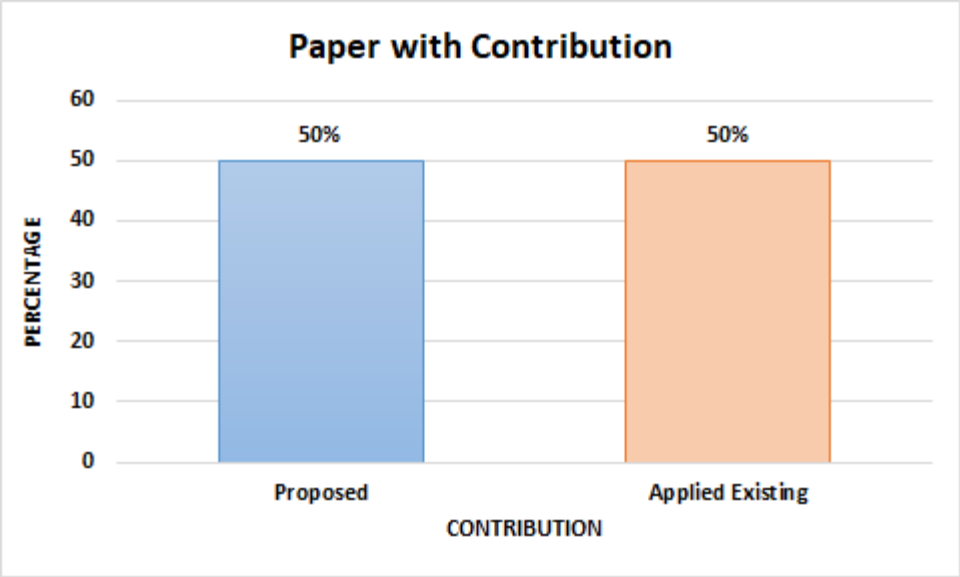


Figure 5: Comparison of papers in which proposed techniques are compared with existing techniques that have been applied.

Level Distribution

A study of the proposed and applied existing papers was undertaken to determine the technique which has been used. In contrast to the findings in the literature review, the outcome of this particular study demonstrates that there are three distinct classifications of techniques for balancing of datasets as shown in Figure 6. The grouping or classification of these techniques are defined as levels to be consistent with published literature presented in Section II.

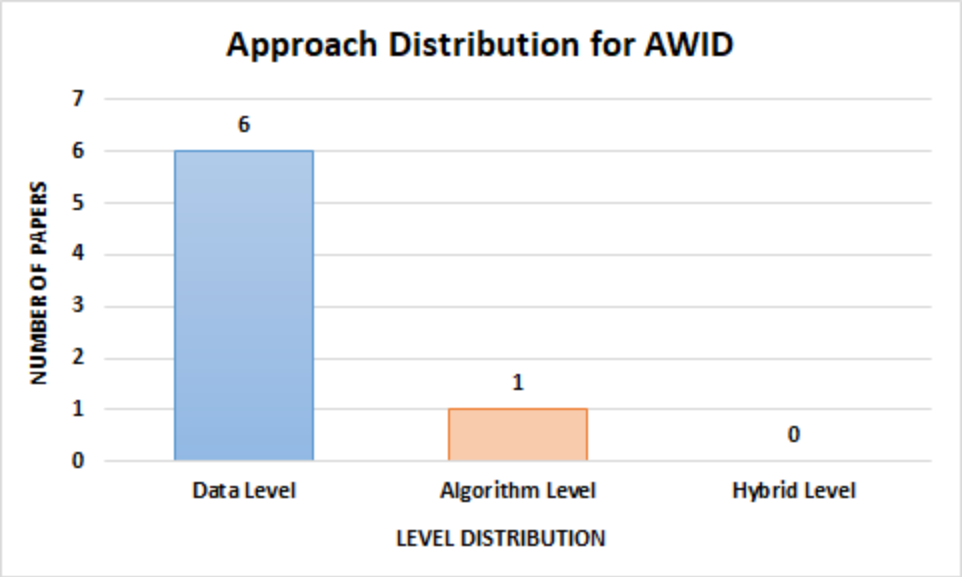


Figure 6: Comparing the number of papers that propose, apply or mention approaches across AWID datasets

The statistical representation in Figure 6 spans the 26 papers published in the IDS dataset. This dataset shows that the majority of papers use data level techniques to solve the issue of imbalance.

3.5.2 Ranking of undertaken approaches

The percentage distribution for ***proposed*** and ***applied existing*** from the total amount of papers ***with contribution*** is depicted in Figure 6. Papers with contribution had 46% of new methods proposed and 54% of existing methods applied. This result may be explained by the fact that a majority of the papers have not focused on proposing a new technique to overcome bias. The two papers identified in Figure 8 account for 5% of the papers ***with contribution*** that fall inside our proposed taxonomy of hybrid level. This discrepancy can be a dominant focus area for future research directions.

Chapter 4

Experimentation Environment

This chapters delivers the outline for implementing and testing the produced framework and utilizing the machine learning approaches.

4.1 Setup

To perform the implementation and testing, Jupyter Notebook was required. It is an open source web application that allows to create and share the document providing live code, equation, visualization and narrative texts. This experiment was performed on the device featuring a Linux (Fedora release 32) HP ProLiant DL380p Gen8. It is a Dual Intel(R) Xenon(R) with CPU E5-2660 v2 @ 2.20GHz (40 cores) and having a memory of 256 GB ECC (1866MT/s) and storage of 9TB.

In particular, Pandas, NumPy and Matplotlib are used as the core packages. Pandas and NumPy libraries are used for loading the data and to perform the preprocessing steps. The NumPy package is an fundamental use for the scientific computing in python. Scikit learn and Matplotlib are used for training and evaluating the model. The Matplotlib library delivers a better quality of figures, as it is used for 2D plotting [40].

4.2 Preprocessing steps

The preprocessing steps are very useful and essential in achieving a proper data to train and test. It is must, that every dataset are applied towards preprocessing. Similarly, the AWID dataset was explored, it also needs certain preprocessing steps to cycle in a proper plan. Thus, the preprocessing procedures were carried on the AWID-CLS-R-Trn subset.

The AWID subset is a single .CSV file that was explored and well understood. According to that, the procedure was followed as below:

1. **Dropping the Empty values\ Empty columns:** The subset file was found to have many of the empty values in the columns. These missing values were considered as “zeros” and was made to drop.

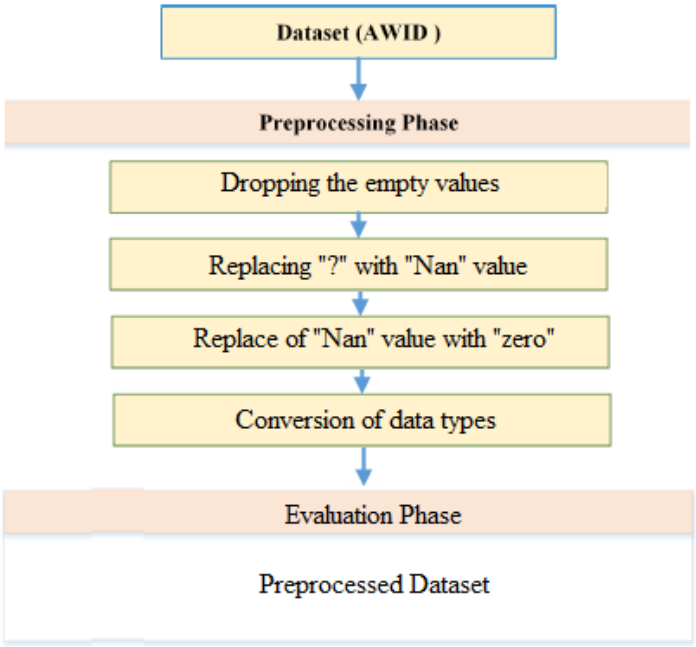
2. **Replace of “?” with “Nan”:** Next, most of the values recorded were represented by the symbol “?”. This symbol were targeted by replacing with “Nan” as a value. Where, it is said to be “Not a number”.

3. **Replace of “Nan” with “0”:** Now, these replaced “Nan” values are reflected as“zeros” in order to remove all the unnecessary values existed.

4. **Conversion of data type:** In this conversion step, there were few attributes which had hexa-decimal and float values with them. These attributes were converted into integer values.

This primary steps in preprocessing delivered the modified dataset with 48 attributes. And hence, it is now proper to test and train in the evaluation process.

Fig 7 Preprocessing steps followed in AWID



4.3 Structure of AWID-CLS-R-Trn

The AWID-CLS-F-Trn subset and AWID-CLS-R-Trn subset have similar types of class distribution with different records observed. The structure of the AWID-CLS-R-Trn is defined as a reduced version of training set which contains four classes. The real traces of both intrusion and normal are observed with 1,795,575 records, in 1 hour. This subset has the type of class distribution consists of Flooding, Impersonation, Injection and Normal. These attributes values are separated by a comma. The extracted file of this subset occupies a size of 844MB and available in the extension of .CSV format [34].

Table 5. The characteristics of AWID-CLS-R-Trn

AWID-CLS-R-Trn	
Attack Type	**Counts**
Flooding	48,484
Impersonation	48,522
Injection	65,379
Normal	1,633,190

Table 5**,** highlights the characteristics of the above considered subset of AWID. The Flooding, Impersonation and Injection are represented as the attack classes and the normal intrusion is said to be normal class. The total number of records in the training set is 1,795,575. From the observations, the normal class encloses a value of 1,633,190 records and the three attack classes have 162,385 records in total.

4.3.1 AWID feature description

The AWID datasets consists of 155 attributes in total. But, in this study, among those attributes 48 were considered and used for the evaluation. The rest of them were dedicated during the pre-processing stage. The 48 attributes which are considered for the evaluation are listed below:

Table.6 The AWID feature description

AWID Features			
1	frame.interface_id	25	radiotap.present.antenna
2	frame.offset_shift	26	radiotap.present.db_antsignal
3	frame.time_epoch	27	radiotap.present.db_antnoise
4	frame.time_delta	28	radiotap.present.rxflags
5	frame.time_delta_displayed	29	radiotap.present.xchannel
6	frame.time_relative	30	radiotap.present.mcs
7	frame.len	31	radiotap.present.ampdu
8	frame.cap_len	32	radiotap.present.vht
9	frame.marked	33	radiotap.present.reserved
10	frame.ignored	34	radiotap.present.rtap_ns
11	radiotap.version	35	radiotap.present.vendor_ns
12	radiotap.pad	36	radiotap.present.ext
13	radiotap.length	37	radiotap.datarate
14	radiotap.present.tsft	38	wlan.fc.type_subtype
15	radiotap.present.flags	39	wlan.fc.version
16	radiotap.present.rate	40	wlan.fc.type
17	radiotap.present.channel	41	wlan.fc.subtype
18	radiotap.present.fhss	42	wlan.fc.ds
19	radiotap.present.dbm_antsignal	43	wlan.fc.frag
20	radiotap.present.dbm_antnoise	44	wlan.fc.retry
21	radiotap.present.lock_quality	45	wlan.fc.pwrmgt
22	radiotap.present.tx_attenuation	46	wlan.fc.moredata
23	radiotap.present.db_tx_attenuation	47	wlan.fc.protected
24	radiotap.present.dbm_tx_power	48	wlan.fc.order

4.3.2 Attacks in AWID

The classification in AWID are comprised of both larger and reduced set of packets. These classification are utilized accordingly when the attack takes place. Among these classification, this work is focused on the AWID-CLS-R-Trn set. The class distribution of AWID-CLS-R-Trn consists of four categories, such as: Flooding, Impersonation, Injection and Normal. Among them, the flooding and the impersonation attacks are used for the implementation.

Flooding Attack

The flooding attack is one of the intrusion which is straight forward to execute, though it cause unsettling influences inside the network. There can be one or more number of intruders try to get into the network. Once they enter the network, the attackers use data flooding concept to easily get interrupted. Thus, a large volumes of data gets infused in order to reduce the network speed. This type of flooding attacks are used mostly prior to DoS attack [41].

Impersonation Attack

The intruder tries to figure out to get into a wireless network without the knowledge, this occurrence is the cause for the impersonation attack. It is very difficult to recognize this event as the framework approves them as the authentic client [42].

4.4 Classifiers

There are different types of classifier models that are being used and developed for many applications. Every available algorithms are different in their characteristics and yet the choice of choosing the model may bring out vast variation in the result. Thus, these supervised machine learning techniques would help in solving the various factors of issues.

In this section, there are three classifier models that were used in the classification process are highlighted. The three classifier models that were selected are: (i) K-Nearest Neighbour (KNN) (ii) Random Forest (RF) and (iii) Logistic Regression (LR).

4.4.1 K-Nearest Neighbor

K Nearest Neighbor is referred as non-parametric learning algorithm as it is in contrast to other supervised learning algorithms. This learning algorithm is used to resolve the complications involved in classification and regression process by considering the closest distance between the input instances [43]. It actually retains the previous instances and then, searches the k closest instance in training set as the predicted output. This case of predicted output, follows to be: (i) In classification: it predicts the majority class among the estimated k nearest neighbors and (ii) In regression: it predicts the average value of its k nearest neighbors, which is considered as the output value.

4.4.2 Random Forest

Random Forest learning algorithm is also predominantly used in resolving the classification and regression complications. This supervised learning algorithm is easy and utilizing it involves decision tree which leads to decision forest to perform its functions. The classification accuracy is estimated according to the trained number of decision tress that were created during the process. And hence, the produced outcome is profoundly favored for its accurate result and fast outcomes with varied data and partial cases [8].

4.4.3 Logistic Regression

The Logistic regression in [44] defines, as a machine learning algorithm which analyses a dataset which has one and more independent variables that decide on outcome. These estimated outcomes are in binary variables, either as 0 or 1. The purpose of this classifier algorithm is to portray the relation between the progression among independent variables and its qualities. Mostly, the outcome is generated by estimating the probability of the default class where 1 represents the default class

4.5 Performance Metrics

In this work, the evaluation performance with respect to AWID subset is carried out by utilizing IDS classifiers. The following metrics that were used in this section are totally based on

the actual and the predicted classes. These two sets of classes include True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN). There are different types of performance metrics which are used in evaluating the performance of a model. But, in this study, certain metrics were selected and used for the evaluation process [45]. These metrics are listed and defined below:

1. **Accuracy:** It is the most spontaneous metric which performs, a ratio between the numbers of correctly predicted observation and the total observations. If a model generates high accuracy, it is considered as the best one. But, the best rate is achieved only when there is a symmetric dataset with identical values of both false positive and false negatives. Thus, to support the variance, even other measures are also considered for the performance.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$$

2. **Precision:** It is defined as the ratio of number of correctly predicted positive values to the total number of predicted positive values. When there is a high precision result, which relates to a low false positive rate.

$$\text{Precision} = \frac{TP}{TP+FP}$$

3. **Recall:** It can be defined as the ratio of correctly predicted positive values to the sum of the predictions in the actual class. When there a high in False Negative, it indicates a low value with recall. Recall is also referred as Sensitivity or True Positive Rate (TPR).

$$\text{Recall} = \frac{TP}{TP+FN}$$

4. **F1 score:** It is a harmonic mean value given by the weighted average of both Precision and Recall. This score is especially more useful, when there is a uneven class distribution in a dataset. And here. It is noticed that, both the false positives and false negatives are taken into consideration.

$$F1-\text{Score} = \frac{2 \times (\text{Recall} \times \text{Precision})}{(\text{Recall} + \text{Precision})}$$

Chapter 5

Evaluation of dataset

In this section, the implementation and evaluation of the AWID dataset are provided in detail with the results. As it is highlighted in previous section, Python programming language was used with the Scikit-Learn 3 machine learning library. This Scikit-Learn is an open souce library which comprises several implementations of the machine learning algorithms. There are certain sections where the dataset is experienced in different phases in the evaluation process. They are, as follows:

5.1 Phase I: Modeling on AWID-CLS-R-Trn dataset

Initially, this phase of modeling the dataset is implemented and executed after loading the dataset into the model, with preprocessing. At this stage, the modeling considers the default samples instances of the AWID-CLS-R-Trn. This process deals with training the model by utilizing Scikit-Learning libraries such as: train and train split functions. Thus, the classifier score for this evaluation are provided respectively below:

Table 7. Accuracy achieved in modeling phase

Attack type	Classifier Type	Classifier Score
Flooding	K Nearest Neighbor	0.9996630345272483
	Random Forest	0.9998234084688253
	Logistic Regression	0.9715056950768803
Impersonation	K Nearest Neighbor	0.999818006540953
	Random Forest	0.9999639616912779
	Logistic Regression	0.9712252124007821

5.2 Phase II: Threshold limit on AWID-CLS-R-Trn dataset

In this phase, the threshold limit is determined by considering the highest breakpoint compared in a batch of ten iterations. The training samples were manually and randomly considered with equal intervals for this observation. There are two separate version observed through: (i) keeping the benign sample constant and using random attack samples and: (ii) keeping the attack sample constant and using random benign samples. The second version of execution is to justify an equivalent flow of threshold limit with the first version. The iterations are focused on two of the AWID-CLS-R-Trn dataset attacks with three classifiers. The breakpoint analysis is compared between the two attacks in all the subsections. They observations are represented in the form of graphs below:

5.2.1 K-Nearest Neighbor Classifier:

Figure 8 Threshold in KNN – Flooding version 1

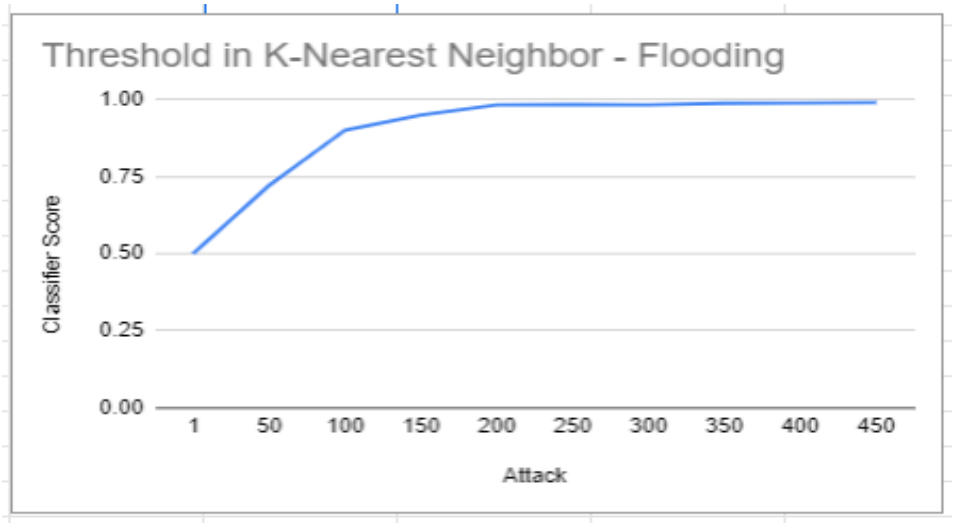
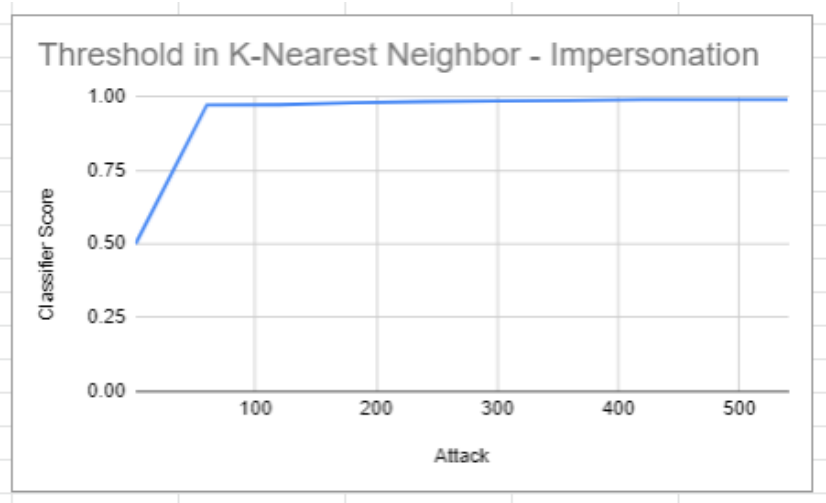


Figure 9 Threshold in KNN – Impersonation version 1



From the figures above, a gradual upward trend can be observed in the Flooding attack, whereas the impersonation attack has an immediate raise. The attack iteration of flooding ranges from 1 to 450 and impersonation has a range of 1 to 540. The lowest accuracy of 50% was observed in both the attack sample at 1. The pre-breakpoint of accuracy was found to be 90% at 100 samples with flooding and 97% at 60 samples with impersonation. Here, it tends to be a sudden increase comparatively. And, the highest was achieved at 450 samples and 420 samples respectively with 99%.

Part 2

This version represents the downfall trend corresponding to their decrease with the benign samples.

Figure 10: Threshold in KNN – Flooding version 2

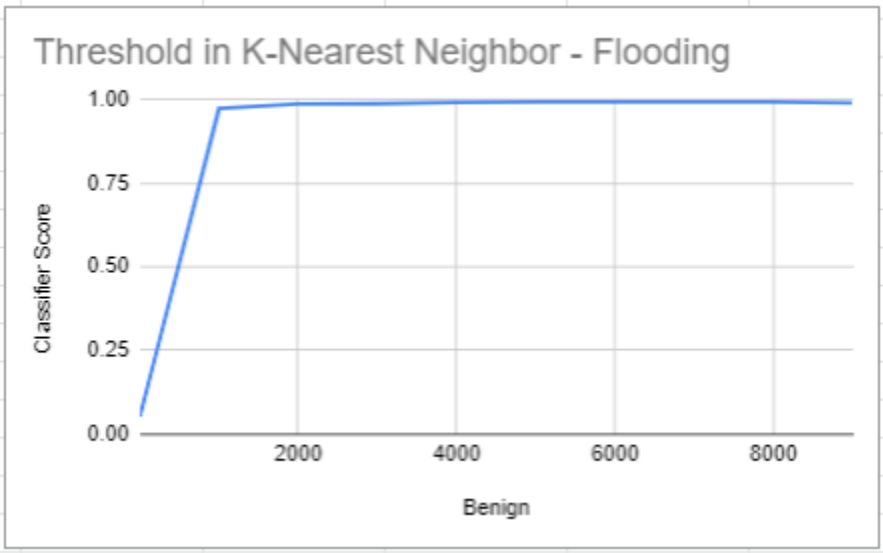
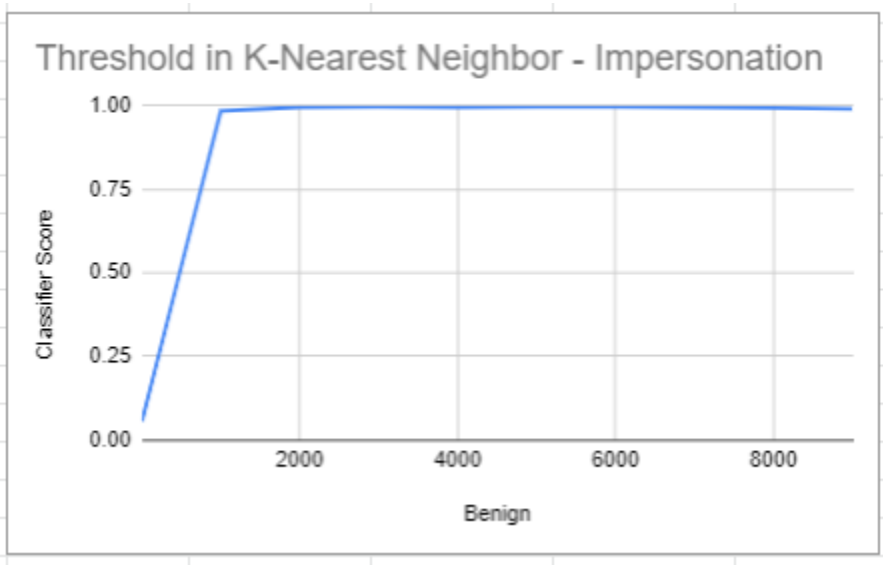


Figure 11: Threshold in KNN – Impersonation version 2



These two attacks share the same type of benign samples and it can be observed that they both have a sudden reducing pattern of accuracy achieved**.** Thus, an equivalent drop was identified on both the attacks from 9000 samples with 99% accuracy reduced to 5% at benign sample 1.

5.2.2 Random Forest Classifier:

Figure 12: Threshold in RF – Flooding version 1

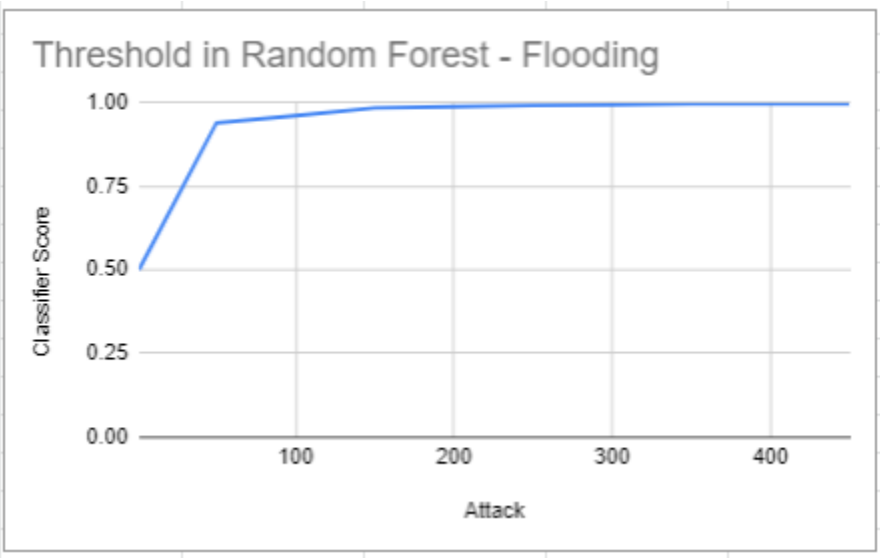
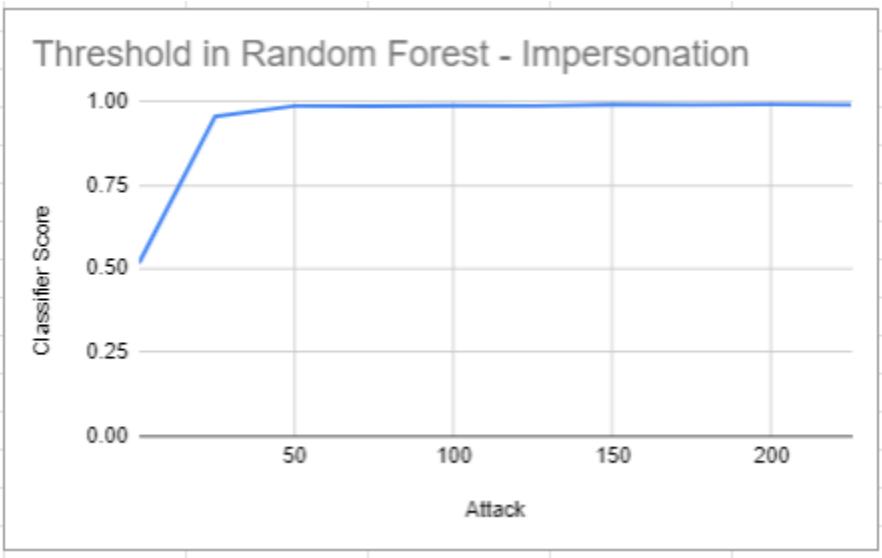


Figure 13: Threshold in RF – Impersonation version 1



An immediate upward trend can be observed from a limit in the both the attacks. The attack iteration of flooding ranges from 1 to 450 and impersonation has a range of 1 to 225. The lowest accuracy of 50% and 51% respectively was observed in attack sample at 1. In this case, the pre-breakpoint of accuracy was found faster as 93% at 50 samples with flooding and 95% at 25 samples with impersonation. Here, it tends to be a sudden increase in common for both the attacks. And, the highest was achieved at 250 samples and 150 samples respectively with 99%.

Part 2

Figure 14: Threshold in RF – Flooding version 2

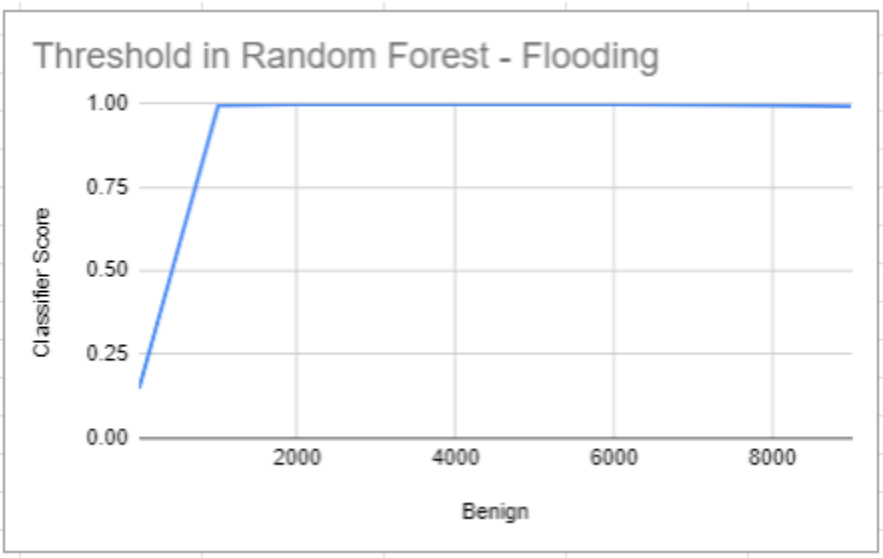
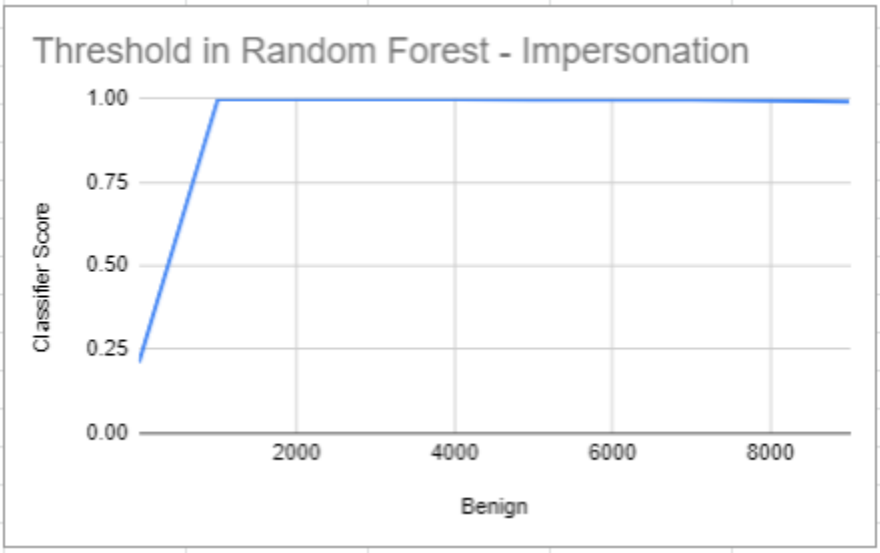


Figure 15: Threshold in RF – Impersonation version 2



These two attacks share the same type of benign samples and it can be observed that they both have a sudden drop pattern in accuracy**.** Thus, an equivalent drop was identified on both the attacks from 9000 samples with 99% accuracy reduced to 14% and 21% respectively at benign sample 1.

5.2.3 Logistic Regression Classifier:

Figure 16: Threshold in LR – Flooding version 1

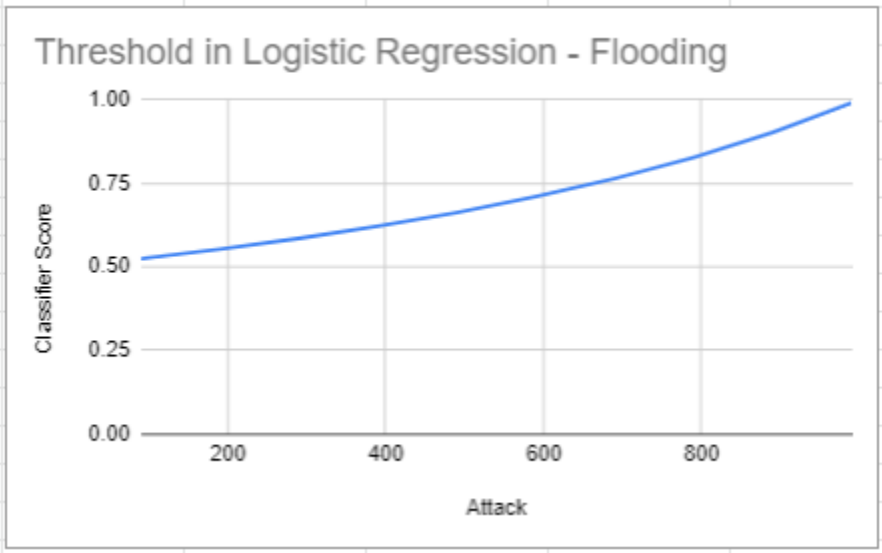
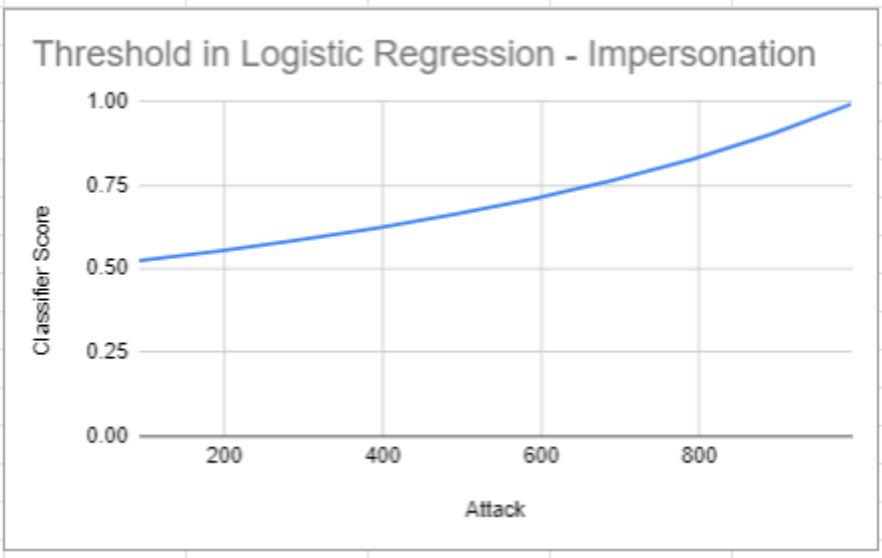


Figure 17: Threshold in LR – Impersonation version 1



A gradual upward curve trend is observed in the both the attacks using logistic regression classifier. The attack iteration of flooding ranges from 90 to 990 and impersonation has a range of 92 to 992. The lowest accuracy of 52% was observed in attack sample at 1, in both the graphs. In this case, the pre-breakpoint of accuracy was found as 90% in both, at 890 samples with flooding and 892 samples with impersonation. And, the highest was achieved at 990 samples and 992 samples respectively with 99%.

Figure 18: Threshold in LR – Flooding version 2

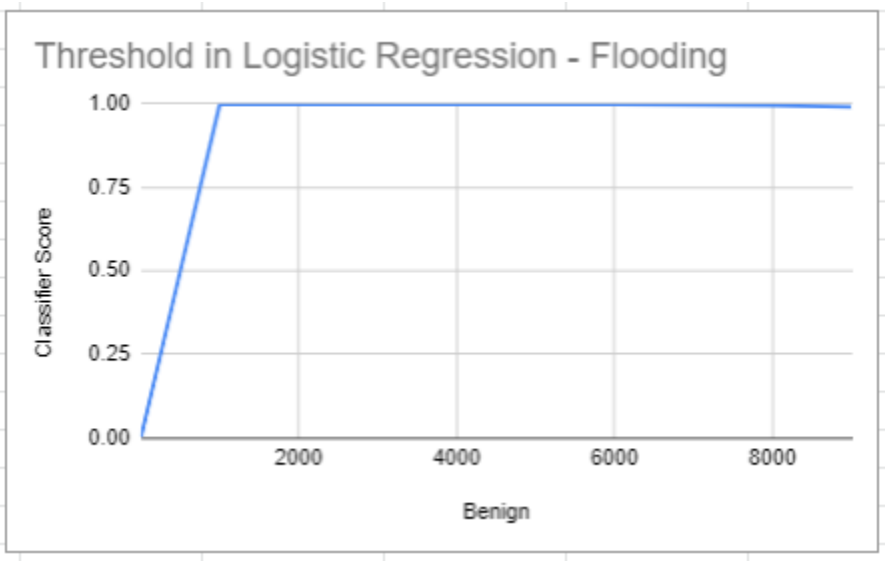
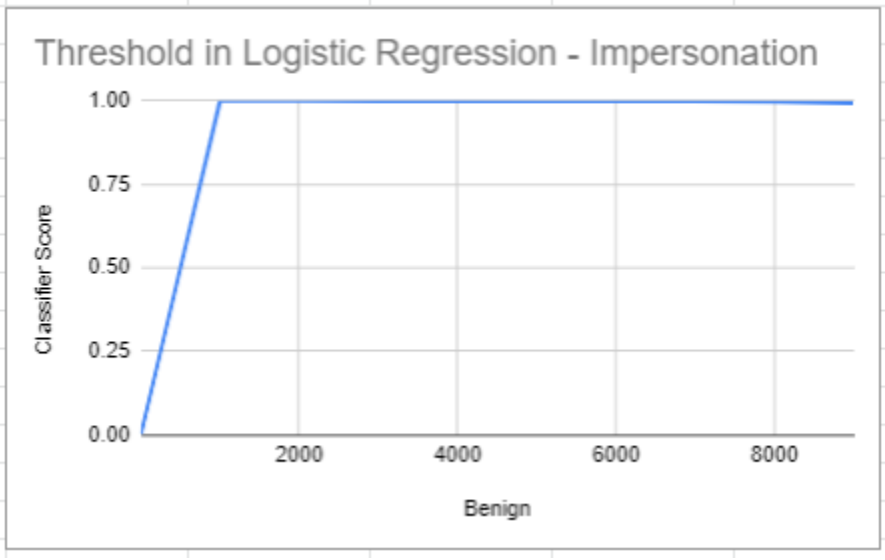


Figure 19: Threshold in LR – Impersonation version 2



These two attacks share the same type of benign samples and it can be observed that they both have a very minute drop pattern in accuracy**.** Even though they had highest accuracy in attack from 9000 samples with 99%, there was an equivalent heavy drop identified on both of the attacks reduced to 0.08% and 0.06% respectively at benign sample 1.

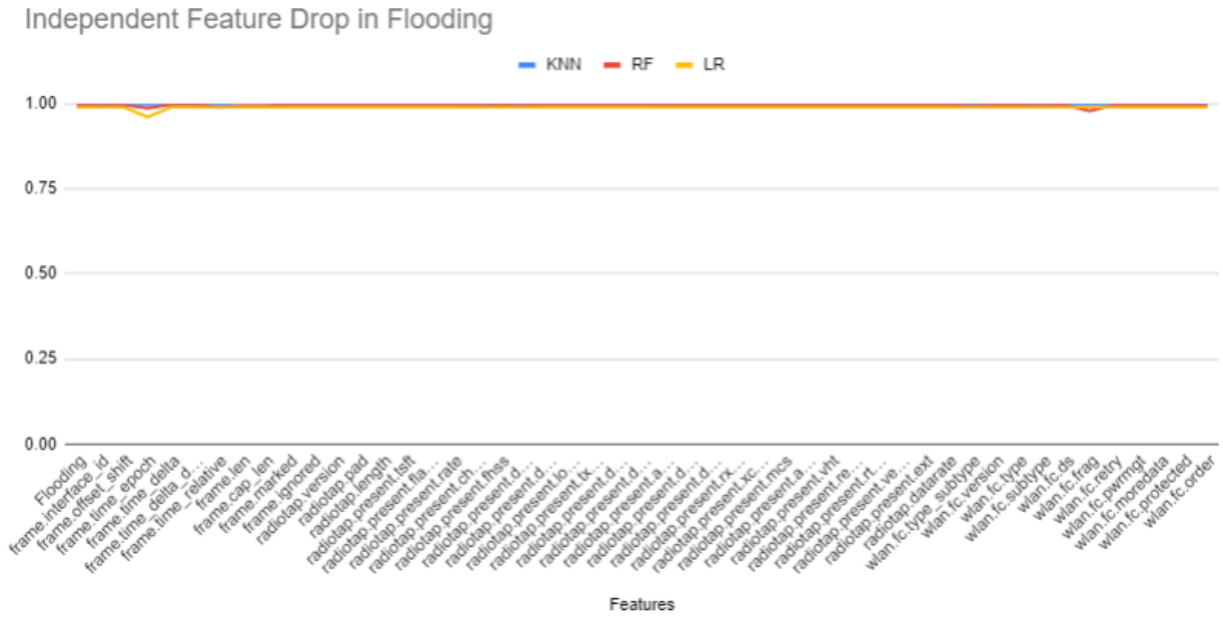
5.3 Phase III: Feature Drop/ Selection on AWID-CLS-R-Trn dataset

In this section, the brake point value for the independent feature drop attempted in flooding and impersonation attacks, obtained from the previous experimentation process. The performance of the three classifier: KNN, RF and LR were used in this feature drop process.

5.3.1 Independent Feature Drop in Flooding Attack

In this section, the accuracy performance of the independent feature drop in flooding attack with three classifiers are discussed below:

Figure 20 Independent Feature Drop in Flooding Attack

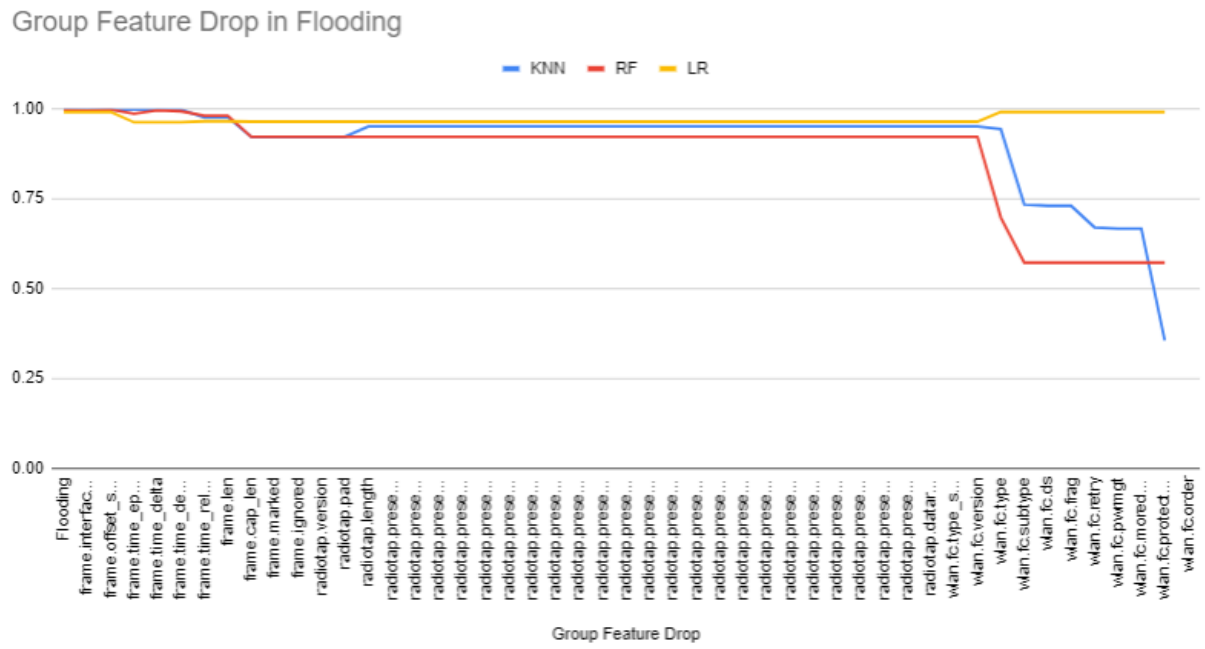


The above graph shows that, in three classifiers recorded the breakpoint values of 99.7%, 99.6% and 99.1% respectively. In KNN, the **frame.len** attribute achieved a drop to 99.3%, as the least accuracy. In RF, **wlan.fc.frag** attribute had a drop to 97.9% and **frame.time_epoch** attribute achieved 96.2% in LR. There were some constant drop among all the three classifiers.

5.3.2 Group Feature Drop in Flooding Attack

In this section, the accuracy performance of the group feature drop in flooding attack with three classifiers are discussed below:

Figure 21 Group Feature Drop in Flooding Attack

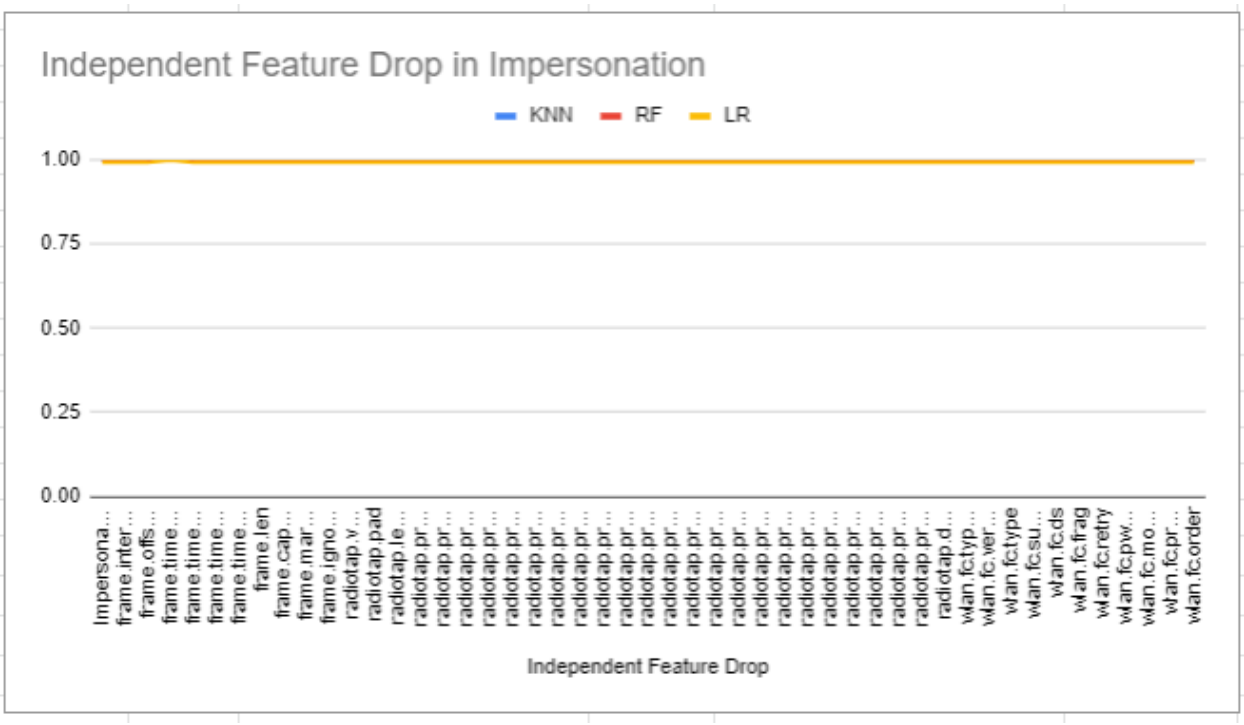


The above graph depicts that, three classifiers have recorded the breakpoint values of 99.7%, 99.6% and 99.1% respectively. The **wlan.fc.protected** attributes was observed to have a downfall with KNN and RF having 35% and 57%. The **frame.time_epoch** attribute in LR achieved a drop to 96.2%. It is observed that there is some constant drop among them.

5.3.3 Independent Feature Drop in Impersonation Attack

In this section, the accuracy performance of the independent feature drop in impersonation attack with three classifiers are discussed below:

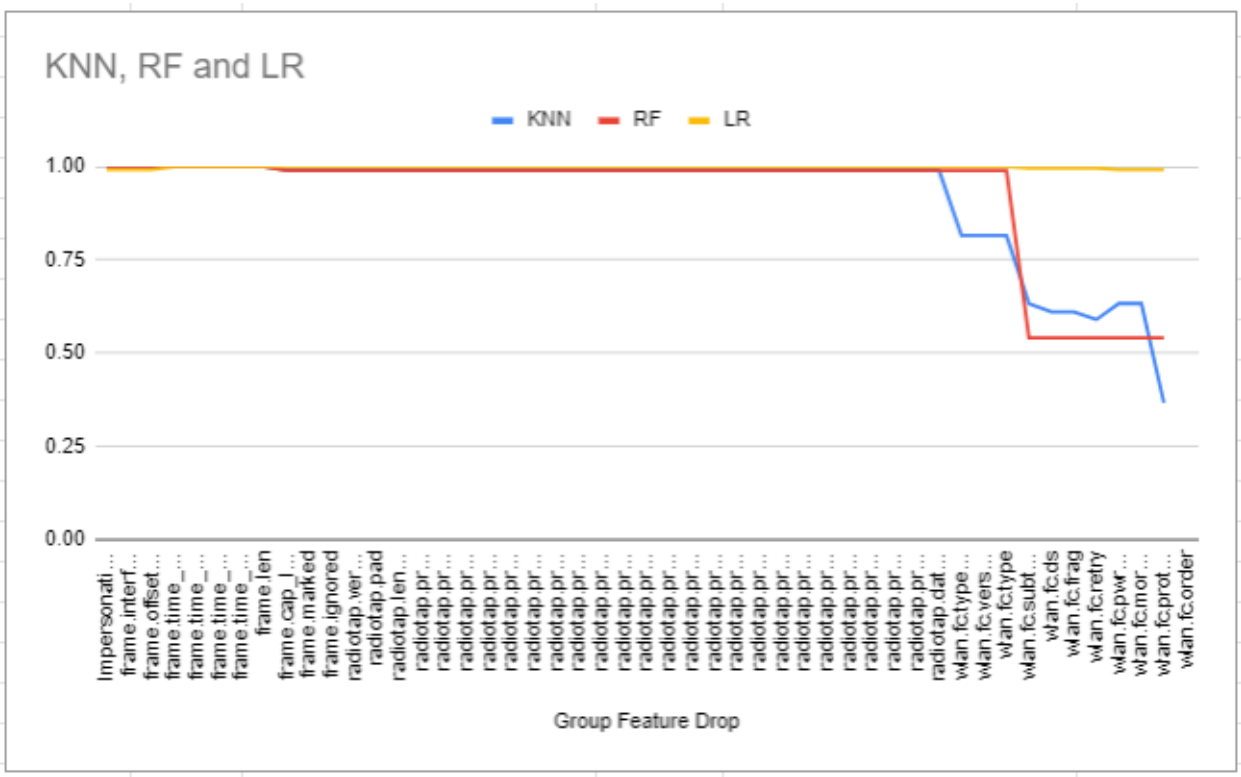
Figure 22 Independent Feature Drop in Impersonation Attack



The above graph depicts that, the classifiers KNN and RF remain constant with the break point accuracy as 100%. The LR has a fluctuation between both low and high and yet, the final drop remained the same as the breaking point as 99.3%. And, then dropping patters was observed constant with the highlighted value.

5.3.4 Group Feature Drop in Impersonation Attack

Figure 23 Group Feature Drop in Impersonation Attack



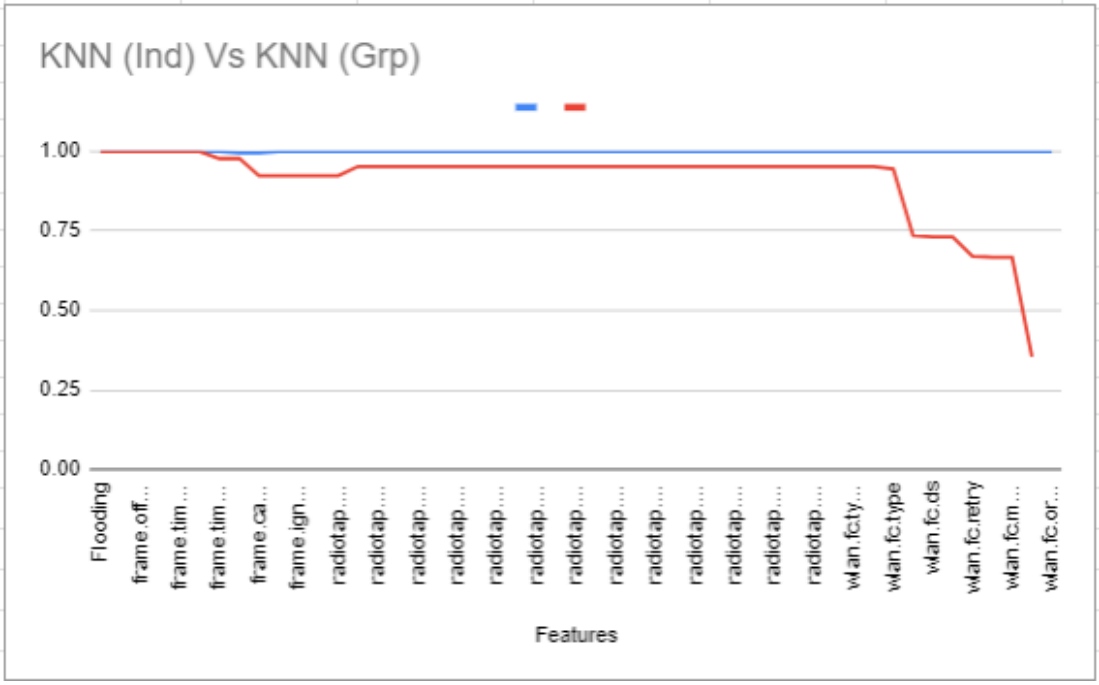
The above graph depicts that, the classifiers KNN and RF remained a constant breakdown values as 1 and LR had obtained 99.3%. The KNN and RF had a small drop exactly on the **frame.cap_len** attribute achieving 99.1% and 98.9%. The **frame.time_epoch** attribute shows an increase in LR and drops to 99% with **wlan.fc.subtype**. And then, the LR has a drop with **wlan.fc.pwrmgt** attribute as 99.3%. It is observed that both the break down value and leat accuracy are same. Thus it kept constant, where as in RF a heavy drop value of 54% and was constant.

5.3.5 Comparison between Feature Drops:

In this section, the both the independent feature drop and group feature drop are compared with the flooding and impersonation attacks. These are discussed on the basis of three classifiers separately. Thus, the observation is derived from the previous section of accuracy performance.

(i) Flooding Attack, KNN in Independent Vs Group:

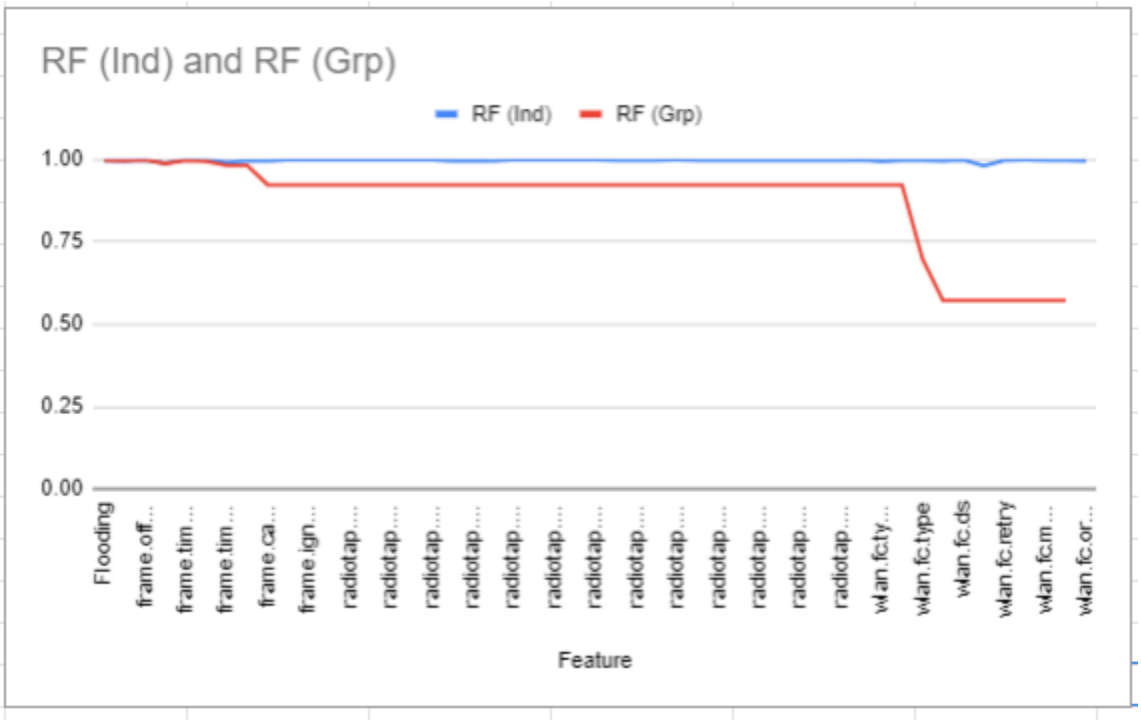
Figure 24 Flooding Attack, KNN in Independent Vs Group



The above graph indicates that the independent feature drop and group feature drop shows a different behavior. Among them, the proper dropping pattern is better with group feature drop in flooding attack using KNN classifier. Thus, the drop was observed in **wlan.fc.protected** attribute with 35% from the break point.

(ii) Flooding Attack, RF in Independent Vs Group:

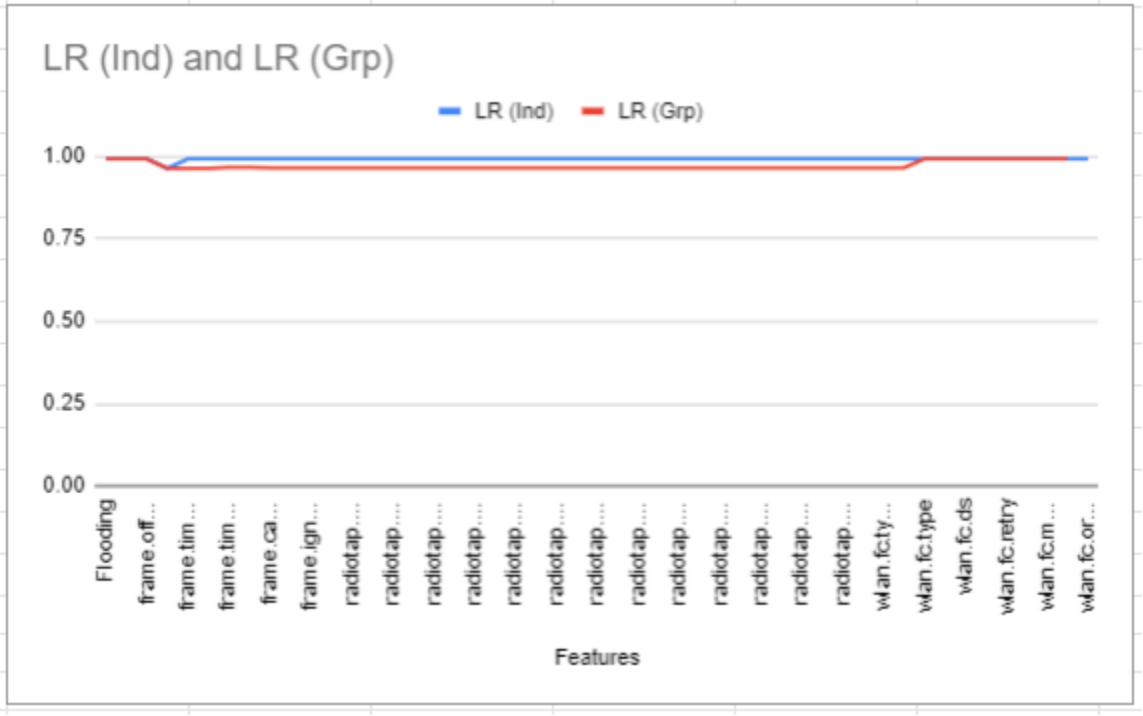
Figure 25 Flooding Attack, RF in Independent Vs Group



The graph depicts that, the RF classifier in group feature drop has a composed dropping pattern. The least accuracy was observed 57% on **wlan.fc.protected** in flooding attack using RF classifier. Whereas, the RF independent feature drop was constant and there was a very low drop observed.

(iii) Flooding Attack, LR in Independent Vs Group:

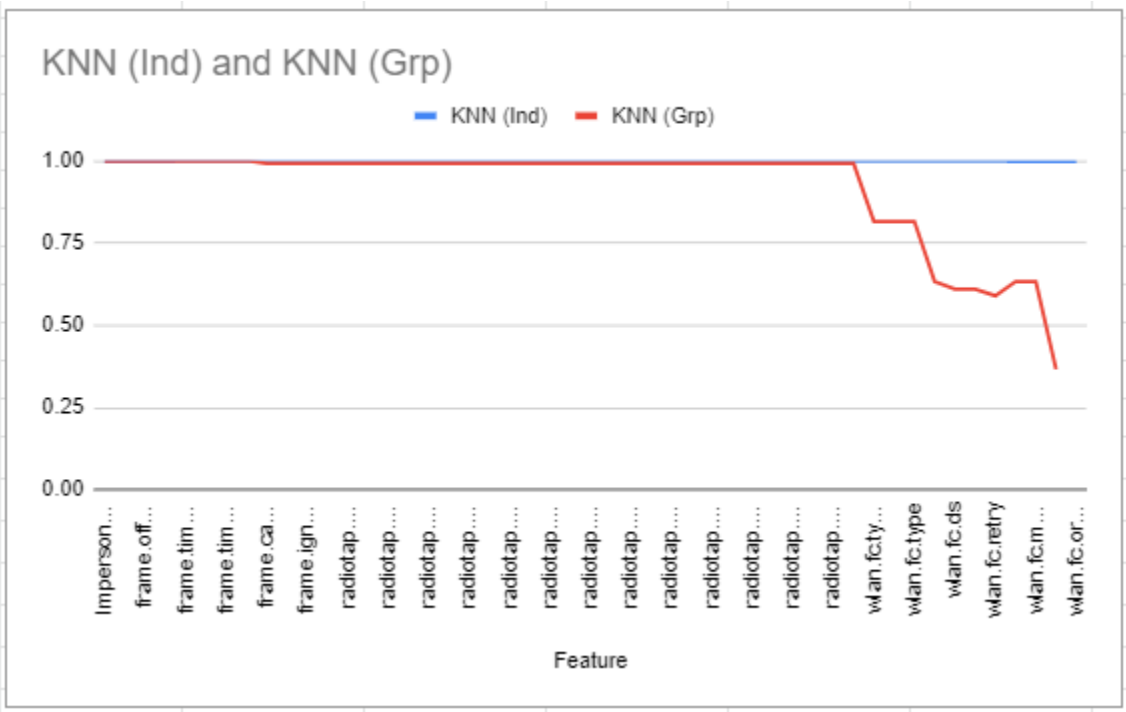
Figure 26 Flooding Attack, LR in Independent Vs Group



It is observed that, both the independent and group feature drop with LR classifier has a fixed state of drop with no difference. There was a common dropping value from 99.1% to 96%. Thus it can be notified that, these have a very minute change in the drop during this process.

(iv) Impersonation Attack, KNN in Independent Vs Group:

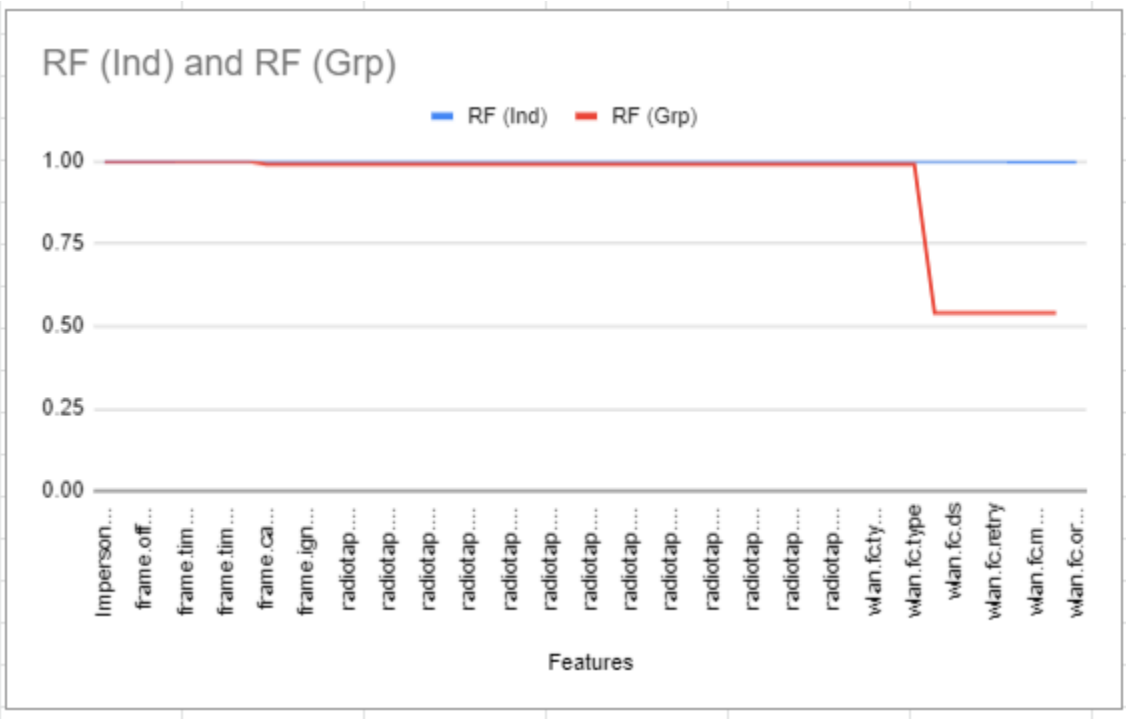
Figure 27 Impersonation Attack, KNN in Independent Vs Group



The above graph indicates that the independent feature drop and group feature drop shows a different behavior. Among them, the proper dropping pattern was observed and is better with group feature drop in impersonation attack using KNN classifier. Thus, the drop was observed in **wlan.fc.protected** attribute with 36% as the least from the break point.

(v) Impersonation Attack, RF in Independent Vs Group:

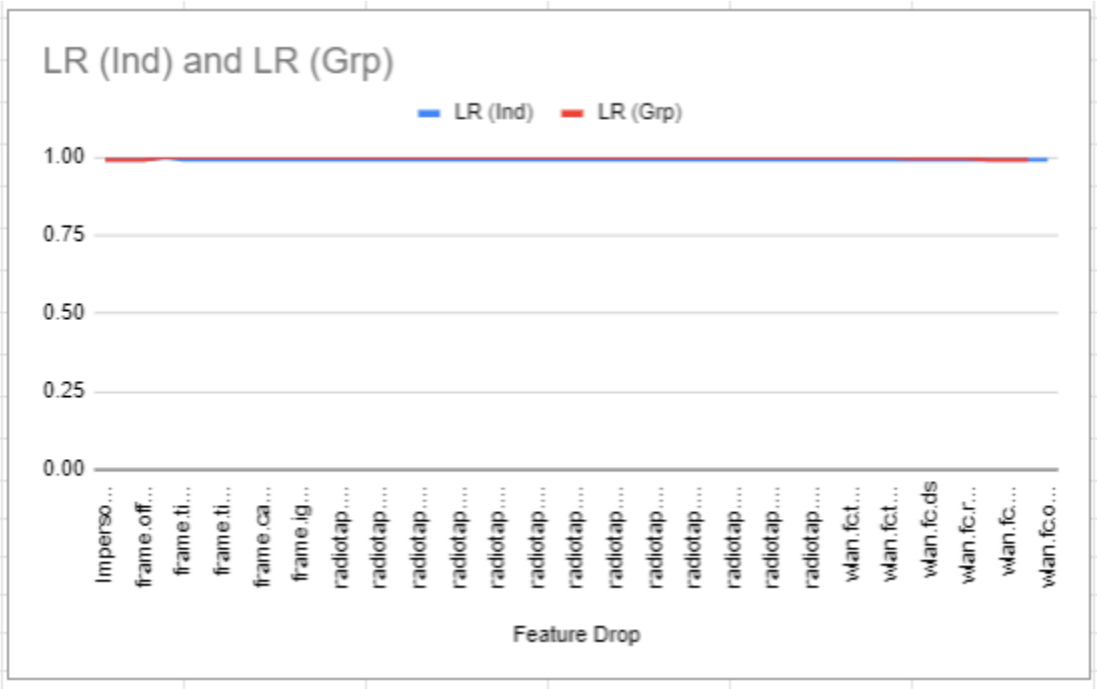
Figure 28 Impersonation Attack, RF in Independent Vs Group



The graph depicts that, the RF classifier in group feature drop has a composed dropping pattern. The least accuracy was observed 54% on **wlan.fc.subtype** in impersonation attack using RF classifier. Whereas, the RF independent feature drop was constant and there was a very low drop observed.

(vi) Impersonation Attack, LR in Independent Vs Group:

Figure 29 Impersonation Attack, LR in Independent Vs Group



It is observed that, both the independent and group feature drop with LR classifier has a fixed state of drop with no difference. The group feature dropping was found to an increase from the break point, but still the accuracy went down to 99.3%. This notifies that, both the break point and the least accuracy value observed from **wlan.fc.pwrmgt** attribute are similar.

In overall, it can be concluded that the behavior of all the classier with two attacks tends to be same. They might have produced a tiny variations among them, but there is nothing as major. Hence, the classifiers of the corresponding feature drop are concluded above representing in graph.

Chapter 6

Conclusion and Future Works

The work demonstrated in this study is associated on the agreement of utilized techniques for balancing the dataset. A proper discussion is provided on the structure of proposed taxonomy involving three levels of techniques in ML. A distributed investigation on the published work and the related work of the utilized dataset encouraged to achieve this compiled output. The aim in bringing up a commitment to both the conclusion and the contribution of this research, would altogether profit towards this study of AWID on IDS for wireless networks. The other majors aspect is that, this study suggested an opportunity to propose a novel technique for handling the imbalanced dataset. The dedicated hybrid level technique can contribute a new combinations of definition. From the investigation of the AWID dataset, it is found that the literatures provide only certain existing parts. The dataset is not fully covered in any of the related researches. Thus, this study tends to be a proposal guideline for the future work which might lead to compare with the existing results. This work assembles, portrays and examines the dataset based on the Wi-Fi Intrusion system. This is a widespread area of research where it is highly depended upon the IEEE 802.11 standards. Based on the resources, the AWID group of datasets are well understood and utilized for the evaluation. The after effects of the assessment have supported keeping up the instances as sensible as the original dataset. Obviously, there might be a chance considered for future scope of AWID committed in advancing its data and growing itself with each forms of the conventions with the 802.11 version of standards.

Bibliography

[1] D. W. F. L. Vilela, E. T. Ferreira, A. A. Shinoda, N. V. De Souza Araujo, R. De Oliveira, and V. E. Nascimento, "A dataset for evaluating intrusion detection systems in IEEE 802.11 wireless networks," *2014 IEEE Colomb. Conf. Commun. Comput. COLCOM 2014 - Conf. Proc.*, no. May 2019, 2014, doi: 10.1109/ColComCon.2014.6860434.

[2] S. Overview, "Cisco Wireless Intrusion Prevention System Configuration Guide ," no. 6387, pp. 1–7, 2013.

[3] A. Tsakountakis, G. Kambourakis, and S. Gritzalis, "Towards effective wireless intrusion detection in IEEE 802.11i," *Proc. - 3rd Int. Work. Secur. Priv. Trust Pervasive Ubiquitous Comput. SecPerU 2007*, no. August 2007, pp. 37–42, 2007, doi: 10.1109/SECPERU.2007.18.

[4] P. Laskov, D. Patrick, and C. Sch, "Learning Intrusion Detection : Supervised or Unsupervised ? Learning intrusion detection : supervised or unsupervised ?," no. September 2005, pp. 50–57, 2014, doi: 10.1007/11553595.

[5] H. Al Najada, I. Mahgoub, and I. Mohammed, "Cyber Intrusion Prediction and Taxonomy System Using Deep Learning and Distributed Big Data Processing," *Proc. 2018 IEEE Symp. Ser. Comput. Intell. SSCI 2018*, pp. 631–638, 2019, doi: 10.1109/SSCI.2018.8628685.

[6] J. L. Leevy, T. M. Khoshgoftaar, R. A. Bauder, and N. Seliya, "A survey on addressing high-class imbalance in big data," *J. Big Data*, vol. 5, no. 1, 2018, doi: 10.1186/s40537-018-0151-6.

[7] A. Ali, S. M. Shamsuddin, and A. L. Ralescu, "Classification with class imbalance problem: A review," *Int. J. Adv. Soft Comput. its Appl.*, vol. 7, no. 3, pp. 176–204, 2015.

[8] G. Karatas, O. Demir, and O. K. Sahingoz, "Increasing the Performance of Machine Learning-Based IDSs on an Imbalanced and Up-to-Date Dataset," *IEEE Access*, vol. 8, pp. 32150–32162, 2020, doi: 10.1109/ACCESS.2020.2973219.

[9] B. Subba, S. Biswas, and S. Karmakar, "A Neural Network based system for Intrusion Detection and attack classification," *2016 22nd Natl. Conf. Commun. NCC 2016*, no. January 2004, 2016, doi: 10.1109/NCC.2016.7561088.

[10] D. Kaleem and K. Ferens, “A cognitive multi-agent model to detect malicious threats,” *Proc. Appl. congigtive Comput. Conf.*, pp. 58–66, 2017.

[11] V. L. L. Thing, “Attack Classification : A Deep Learning Approach,” *2017 IEEE Wirel. Commun. Netw. Conf.*, pp. 1–6, 2017.

[12] C. Kolias, G. Kambourakis, A. Stavrou, and S. Gritzalis, “Intrusion detection in 802.11 networks: Empirical evaluation of threats and a public dataset,” *IEEE Commun. Surv. Tutorials*, vol. 18, no. 1, pp. 184–208, 2016, doi: 10.1109/COMST.2015.2402161.

[13] M. E. Aminanto and K. Kim, “Detecting Active Attacks in WiFi Network by Semi-supervised Deep Learning,” pp. 1–4, 2016.

[14] N. Moustafa and J. Slay, “The significant features of the UNSW-NB15 and the KDD99 data sets for Network Intrusion Detection Systems,” *Proc. - 2015 4th Int. Work. Build. Anal. Datasets Gather. Exp. Returns Secur. BADGERS 2015*, pp. 25–31, 2017, doi: 10.1109/BADGERS.2015.14.

[15] M. E. Aminanto, P. D. Yoo, H. C. Tanuwidjaja, and K. Kim, “Weighted Feature Selection Techniques for Detecting Impersonation Attack in Wi-Fi Networks,” *Symp. Cryptogr. Inf. Secur.*, pp. 1–8, 2017.

[16] U. S. K. P. M. Thanthrige, J. Samarabandu, and X. Wang, “Machine learning techniques for intrusion detection on public dataset,” *Can. Conf. Electr. Comput. Eng.*, vol. 2016-Octob, pp. 7–10, 2016, doi: 10.1109/CCECE.2016.7726677.

[17] U. Sampath, K. Perera, and M. Thanthrige, “Scholarship @ Western,” 2016.

[18] C. Kolias, V. Kolias, and G. Kambourakis, “TermID: a distributed swarm intelligence-based approach for wireless intrusion detection,” *Int. J. Inf. Secur.*, vol. 16, no. 4, pp. 401–416, 2017, doi: 10.1007/s10207-016-0335-z.

[19] F. D. Vaca and Q. Niyaz, “An ensemble learning based Wi-Fi network intrusion detection system (WNIDS),” *NCA 2018 - 2018 IEEE 17th Int. Symp. Netw. Comput. Appl.*, 2018, doi: 10.1109/NCA.2018.8548315.

[20] J. Ran, Y. Ji, and B. Tang, “A semi-supervised learning approach to IEEE 802.11 network anomaly detection,” *IEEE Veh. Technol. Conf.*, vol. 2019-April, 2019, doi: 10.1109/VTCSpring.2019.8746576.

[21] F. Provost and T. Fawcett, “Robust classification for imprecise environments,” *Mach. Learn.*, vol. 42, no. 3, pp. 203–231, 2001, doi: 10.1023/A:1007601015854.

[22] J. M. Johnson and T. M. Khoshgoftaar, “Survey on deep learning with class imbalance,” *J. Big Data*, vol. 6, no. 1, 2019, doi: 10.1186/s40537-019-0192-5.

[23] “Imbalanced Data : How to handle Imbalanced Classification Problems”. [Online]. Available: https://www.analyticsvidhya.com/blog/2017/03/imbalanced-data-classification/

[24] “Ensemble methods: bagging, boosting and stacking | by Joseph ...”. [Online]. Available: https://towardsdatascience.com/ensemble-methods-bagging-boosting-and-stacking-c9214a10a205.

[25] Ling C.X., Sheng V.S. (2011) Cost-Sensitive Learning. In: Sammut C., Webb G.I. (eds) Encyclopedia of Machine Learning. Springer, Boston, MA. https://doi.org/10.1007/978-0-387-30164-8_181

[26] V. Miškovic, “Machine Learning of Hybrid Classification Models for Decision Support,” no. June, pp. 318–323, 2014, doi: 10.15308/sinteza-2014-318-323.

[27] O. Elezaj, S. Y. Yayilgan, M. Abomhara, P. Yeng, and J. Ahmed, “Data-driven intrusion detection system for small and medium enterprises,” *IEEE Int. Work. Comput. Aided Model. Des. Commun. Links Networks, CAMAD*, vol. 2019-Septe, 2019, doi: 10.1109/CAMAD.2019.8858166.

[28] Y. Zhou, G. Cheng, S. Jiang, and M. Dai, “Building an efficient intrusion detection system based on feature selection and ensemble classifier,” *Comput. Networks*, vol. 174, no. October 2019, 2020, doi: 10.1016/j.comnet.2020.107247.

[29] J. W. Mikhail, J. M. Fossaceca, and R. Iammartino, “A semi-boosted nested model with sensitivity-based weighted binarization for multi-domain network intrusion detection,” *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 3, 2019, doi: 10.1145/3313778.

[30] Joseph W . Mikhail, “(PhD" Good intro for Feature Selection) An Investigation of Anomaly-based Ensemble Models for Multi-Domain Intrusion Detection ///Hybrid Naive Bayes Decision Tree Ensemble,” no. January 2010, 2019.

[31] R. Abdulhammed, “Intrusion Detection: Embedded Software Machine Learning and Hardware Rules Based Co-Designs,” p. 176, 2019

[32] Y. Sun, A. K. C. Wong, and M. S. Kamel, “Classification of imbalanced data: A review,” *Int. J. Pattern Recognit. Artif. Intell.*, vol. 23, no. 4, pp. 687–719, 2009, doi: 10.1142/S0218001409007326.

[33] M. H. Abdulraheem and N. B. Ibraheem, “A detailed analysis of new intrusion detection dataset,” *J. Theor. Appl. Inf. Technol.*, vol. 97, no. 17, pp. 4519–4537, 2019.

[34] “Description - Wireless Security Datasets Project”. [Online]. Available: http://icsdweb.aegean.gr/awid/features.html.

[35] “A Primer to Ensemble Learning – Bagging and Boosting”. [Online]. Available: https://analyticsindiamag.com/primer-ensemble-learning-bagging-boosting/.

[36] “Development and Evaluation of a Deep Learning Based,” vol. 0002, no. August, 2017.

[37] S. Rezvy, Y. Luo, M. Petridis, A. Lasebae, and T. Zebin, “An efficient deep learning model for intrusion classification and prediction in 5G and IoT networks,” *2019 53rd Annu. Conf. Inf. Sci. Syst. CISS 2019*, 2019, doi: 10.1109/CISS.2019.8693059.

[38] L. McNabb and R. S. Laramee, “How to Write a Visualization Survey Paper : A Starting Point,” 2019, doi: 10.2312/eged.20191026.

[39] J. Beel, B. Gipp, and E. Wilde, “Academic search engine optimization (ASEO/span),” *J. Sch. Publ.*, vol. 41, no. 2, pp. 176–190, 2010, doi: 10.3138/jsp.41.2.176.

[40] P. Lavanya, A. Sangeetha, and S. Krishnan, “Intrusion detection using machine learning,” *Int. J. Recent Technol. Eng.*, vol. 8, no. 2 Special Issue 6, pp. 832–837, 2019, doi: 10.35940/ijrte.B1154.0782S619.

[41] T. Shang and L. Y. Gui, “Identification and prevention of impersonation attack based on a new flag byte,” *Proc. 2015 4th Int. Conf. Comput. Sci. Netw. Technol. ICCSNT 2015*, no. Iccsnt, pp. 972–976, 2016, doi: 10.1109/ICCSNT.2015.7490899.

[42] E. Alomari, S. Manickam, B. B. Gupta, S. Karuppayah, and R. Alfaris, “Botnet-based Distributed Denial of Service (DDoS) Attacks on Web Servers: Classification and Art,” *Int. J. Comput. Appl.*, vol. 49, no. 7, pp. 24–32, 2012, doi: 10.5120/7640-0724.

[43] “k-Nearest Neighbor Algorithm - Wiley Online Library”. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/9781118874059.ch7.

[44] “Logistic Regression For Machine Learning”. [Online]. Available: https://machinelearningmastery.com/logistic-regression-for-machine-learning/

[45] E. Solutions and * Name, “Accuracy, Precision, Recall & F1 Score: Interpretation of Performance Measures,” Exsilio Blog, 11-Nov-2016. [Online]. Available:

https://blog.exsilio.com/all/accuracy-precision-recall-f1-score-interpretation-of-performance-measures/#:~:text=F1 score - F1 Score is,and false negatives into account.